



US009147200B2

(12) **United States Patent**
Rajkumar et al.

(10) **Patent No.:** **US 9,147,200 B2**
(45) **Date of Patent:** ***Sep. 29, 2015**

(54) **FREQUENCY CAPPING OF CONTENT
ACROSS MULTIPLE DEVICES**

(56) **References Cited**

(71) Applicant: **Google Inc.**, Mountain View, CA (US)

5,408,950 A 4/1995 Porto
5,892,900 A 4/1999 Ginter et al.

(72) Inventors: **Nareshkumar Rajkumar**, San Jose, CA (US); **Vinod Kumar Ramachandran**, Sunnyvale, CA (US)

(Continued)

(73) Assignee: **Google Inc.**, Mountain View, CA (US)

FOREIGN PATENT DOCUMENTS

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

EP 2270741 1/2011
JP 2004070441 3/2004

(Continued)

This patent is subject to a terminal disclaimer.

OTHER PUBLICATIONS

(21) Appl. No.: **14/629,158**

Ebbert, J., "Is Audience Buying Possible in Mobile Advertising?," AdExchanger.com, Aug. 3, 2011 [online] [Retrieved on Dec. 2, 2011]; Retrieved from the Internet URL: <http://www.adexchanger.com/mobile/audience-buying/>; 9 pages.

(Continued)

(22) Filed: **Feb. 23, 2015**

(65) **Prior Publication Data**

US 2015/0170200 A1 Jun. 18, 2015

Primary Examiner — Khanh Dinh

(74) *Attorney, Agent, or Firm* — Fish & Richardson P.C.

Related U.S. Application Data

(63) Continuation of application No. 13/539,123, filed on Jun. 29, 2012, now Pat. No. 8,966,043, which is a continuation-in-part of application No. 13/458,124, filed on Apr. 27, 2012, now Pat. No. 8,688,984.

(51) **Int. Cl.**

G06F 15/16 (2006.01)

G06Q 30/02 (2012.01)

(Continued)

(52) **U.S. Cl.**

CPC **G06Q 30/0251** (2013.01); **H04L 29/06** (2013.01); **H04L 29/08072** (2013.01)

(58) **Field of Classification Search**

CPC H04L 29/06; H04L 29/08072

USPC 709/220, 223, 224, 228

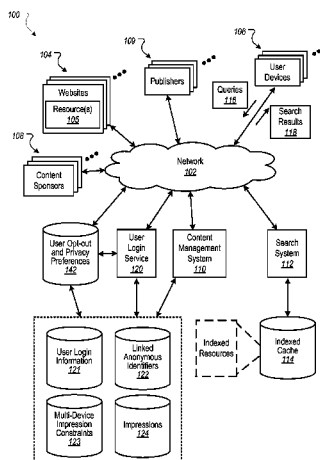
See application file for complete search history.

(57)

ABSTRACT

Methods, systems, and apparatus, including computer programs encoded on a computer-readable storage medium, and including a method for delivering content. The method comprises identifying impressions of content to a user accessing resources using different requesting sources. The method further comprises storing impression data for the identified impressions in association with the user and requesting source. The method further comprises storing requesting source characteristic information with the impression data and identifying parameters that require limits on a number of impressions that are to occur in a time period and type of requesting source. The method further comprises receiving a request for content from a particular requesting source associated with the user, and determining when impressions available for that type of requesting source have been satisfied, and when not, enabling delivery of a content item associated with a campaign to the requesting source responsive to the received request.

12 Claims, 13 Drawing Sheets



- * cited by examiner

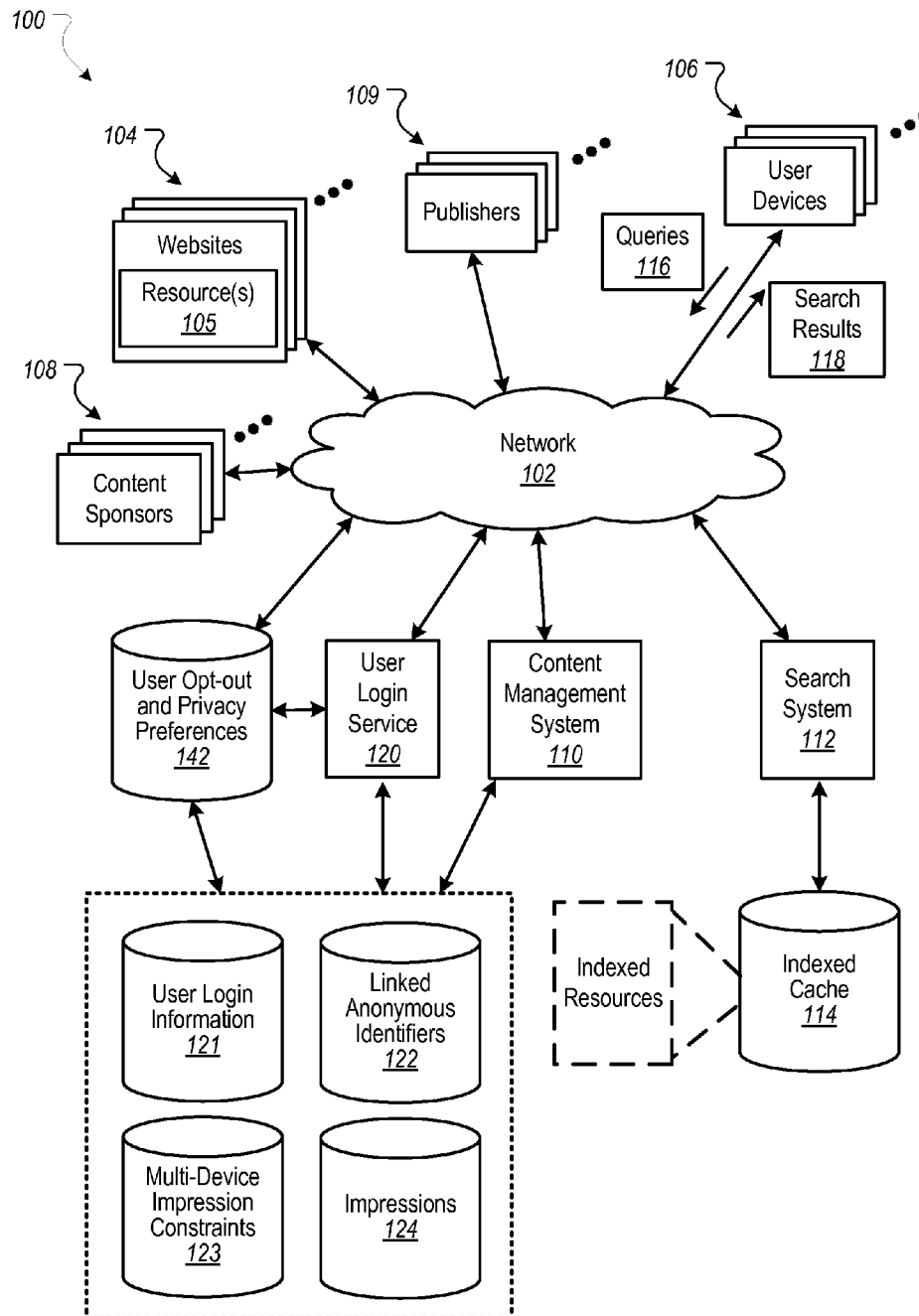


FIG. 1

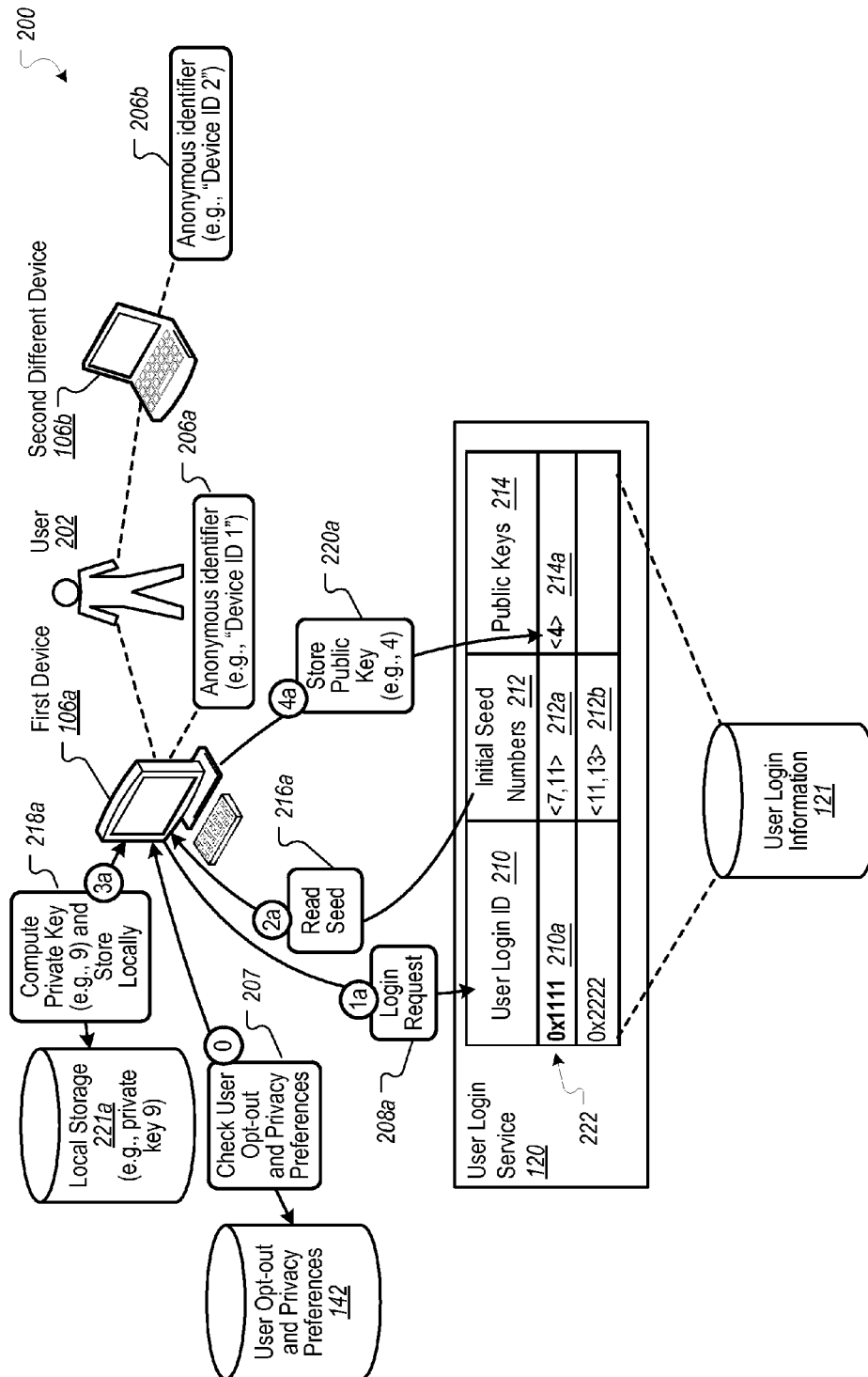


FIG. 2A

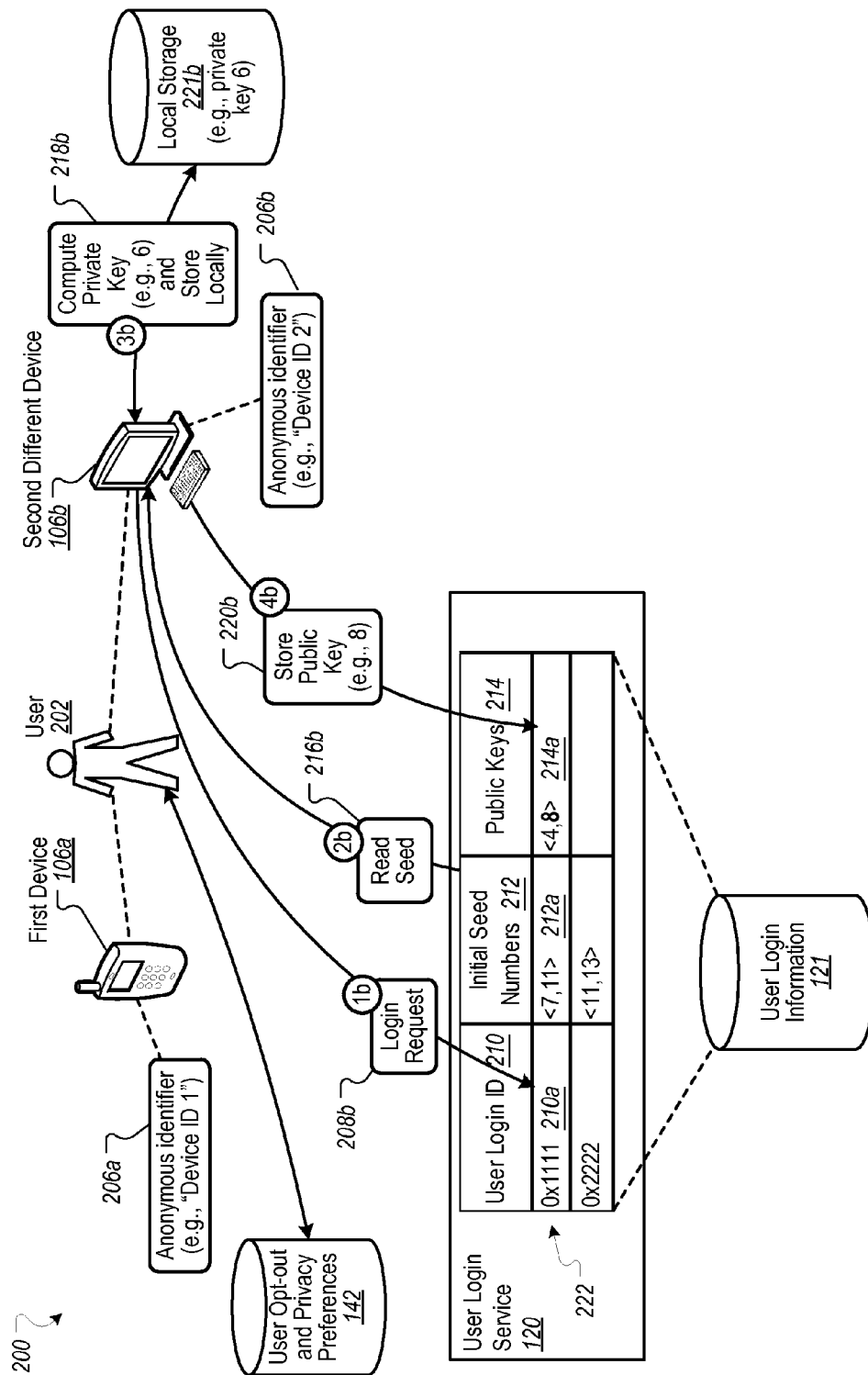


FIG. 2B

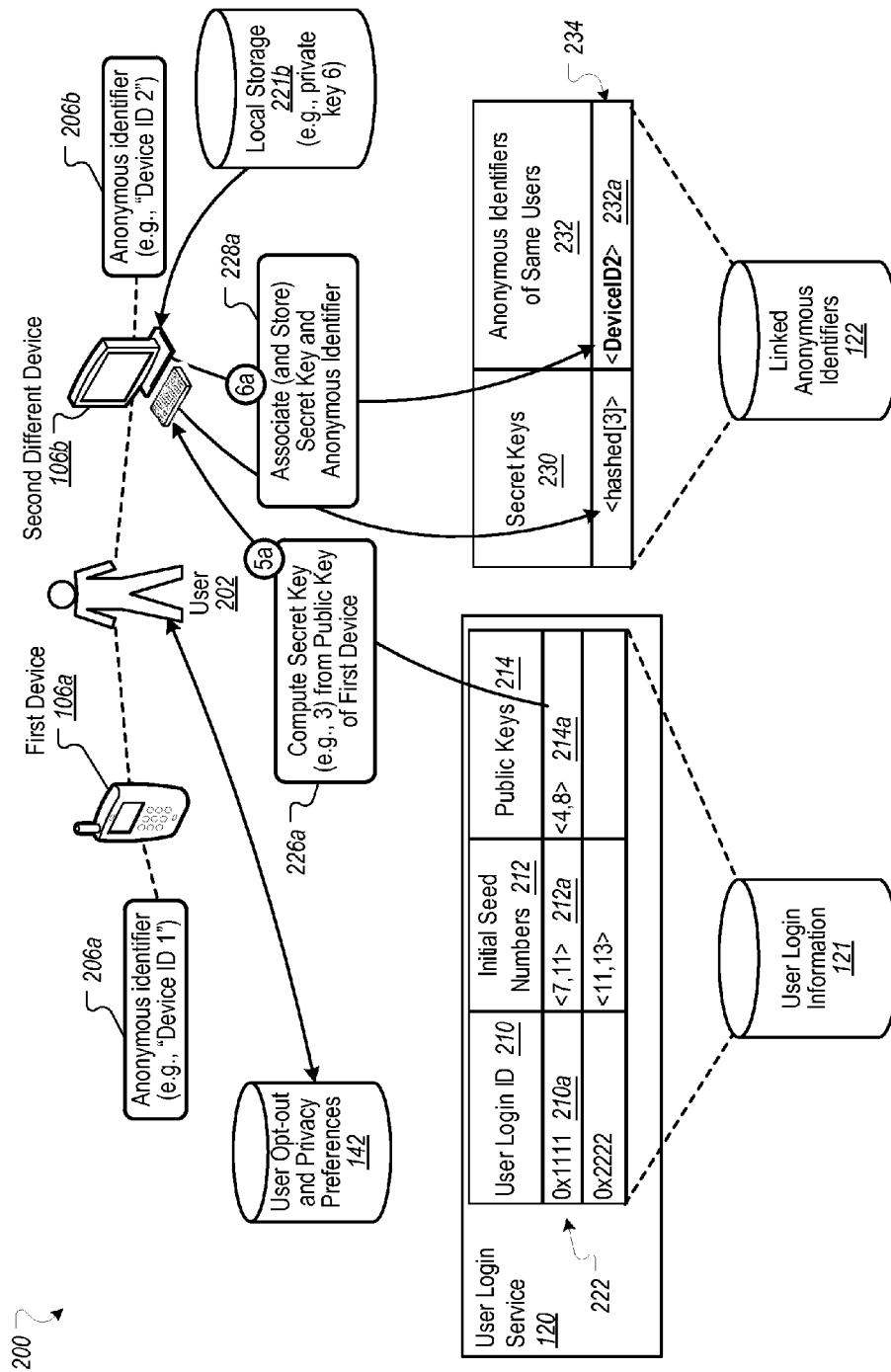


FIG. 2C

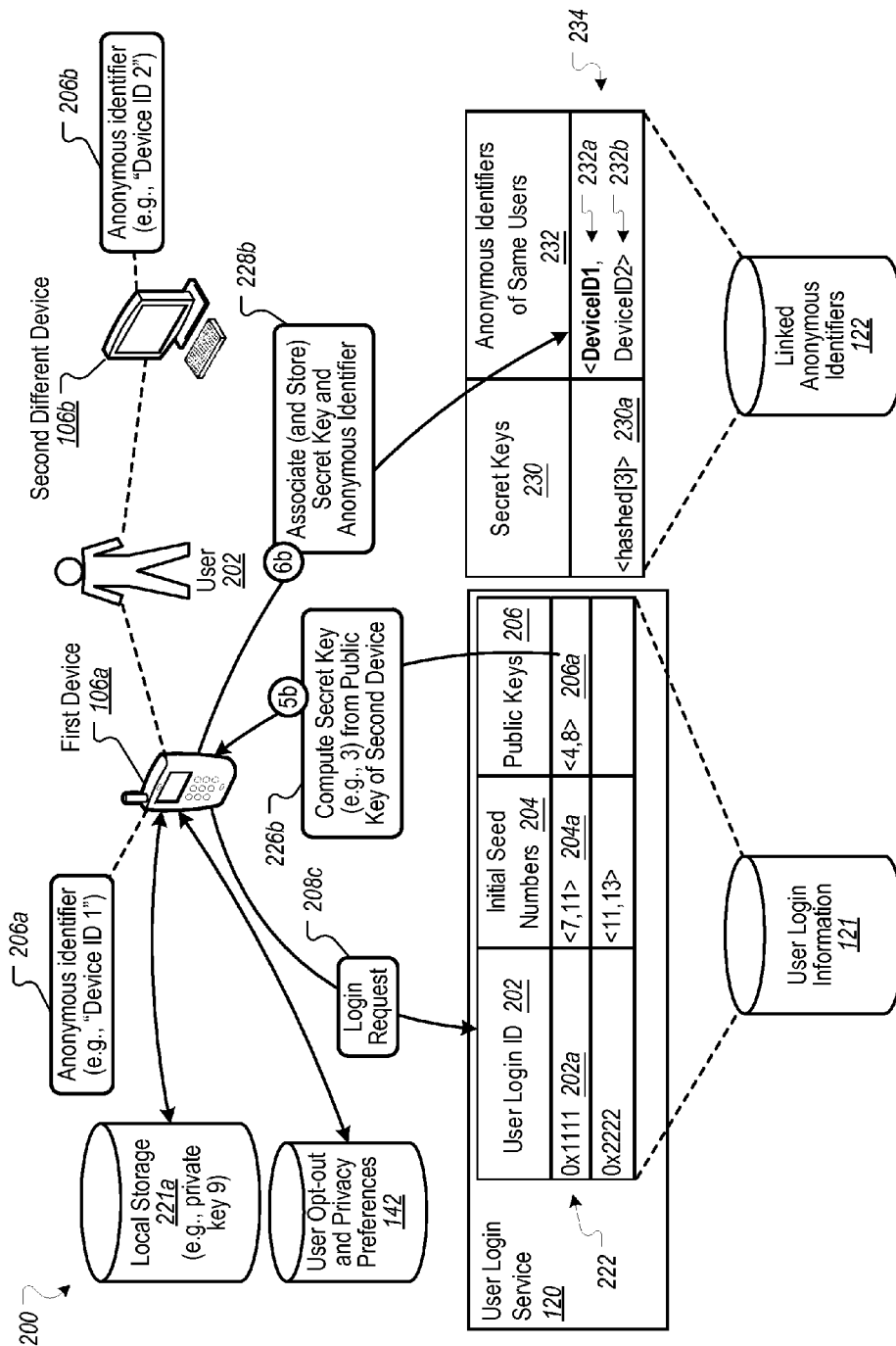


FIG. 2D

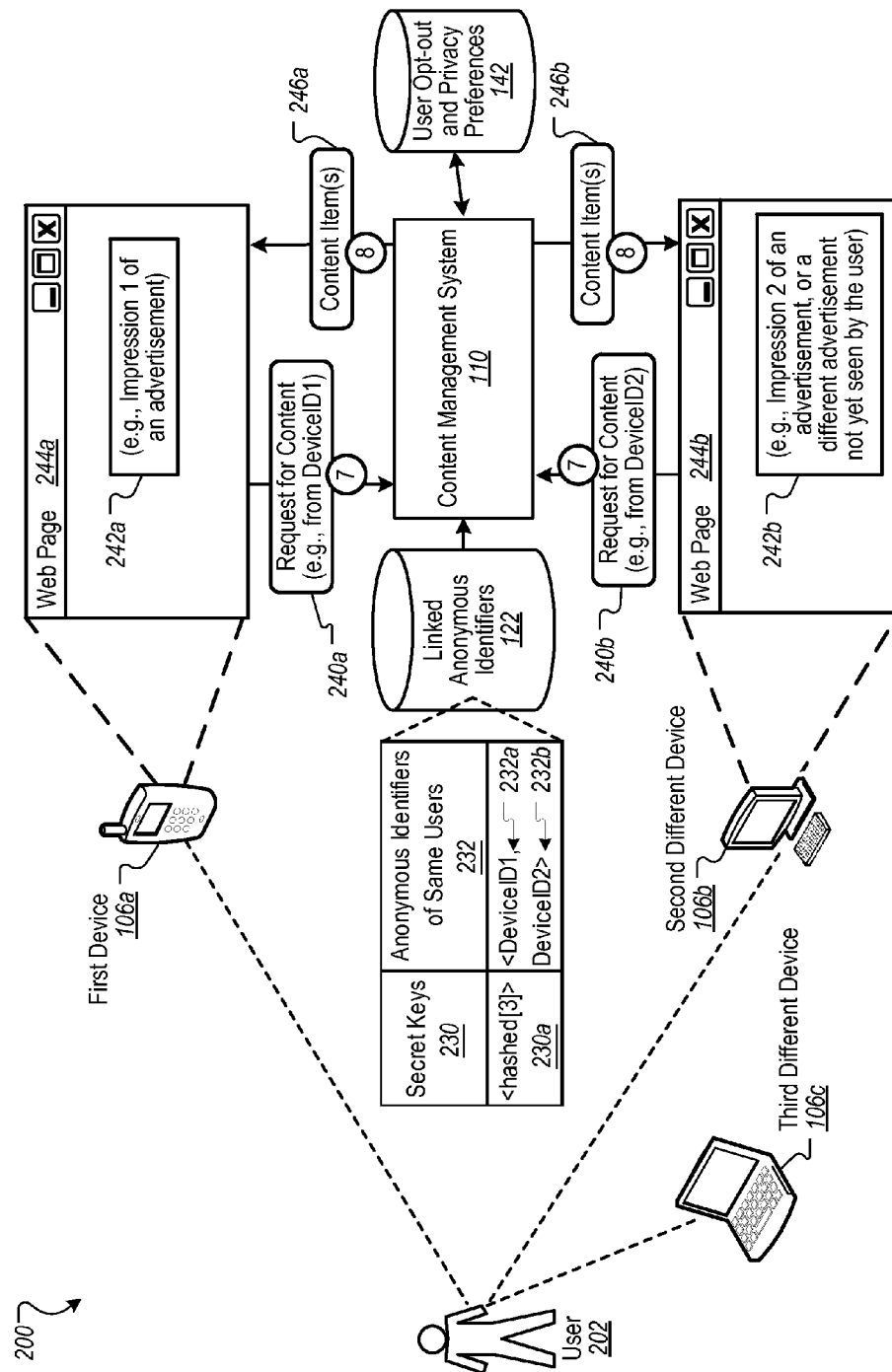


FIG. 2E

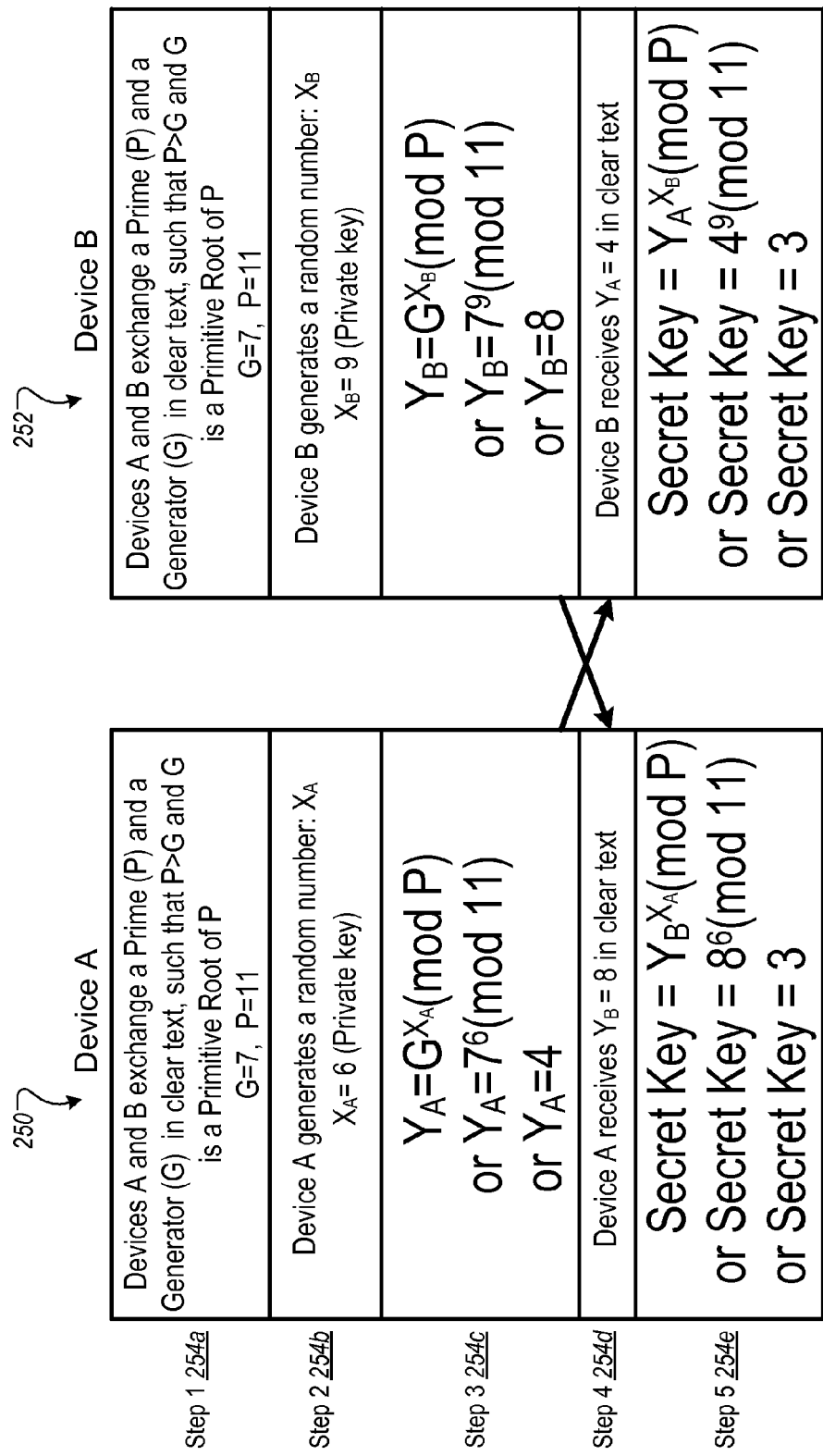


FIG. 2F

Content Sponsor Interface 256 X Another Page X (+)

http://www.example.com/campaigninterface.htm

Campaign Settings

Creative X for Camera X

Multiple-Device Frequency Capping Settings

☒ Set total number of impressions across linked devices 264
Up to total impression(s) on a user's linked devices 262a
Up to impression(s) on a user's linked mobile devices 262b
Up to impression(s) on a user's linked non-mobile devices 262c

☒ End Impressions Upon Conversion 292c
Specify Conversion Events for this Campaign 266

Time Constraints 268

☒ Set patterns of impressions across linked devices 274
Initially, provide impression(s) on a user's linked mobile devices 270a
... followed by impression(s) on a user's linked non-mobile devices 270b
More... 272

Save Settings

Cancel Changes

FIG. 2G

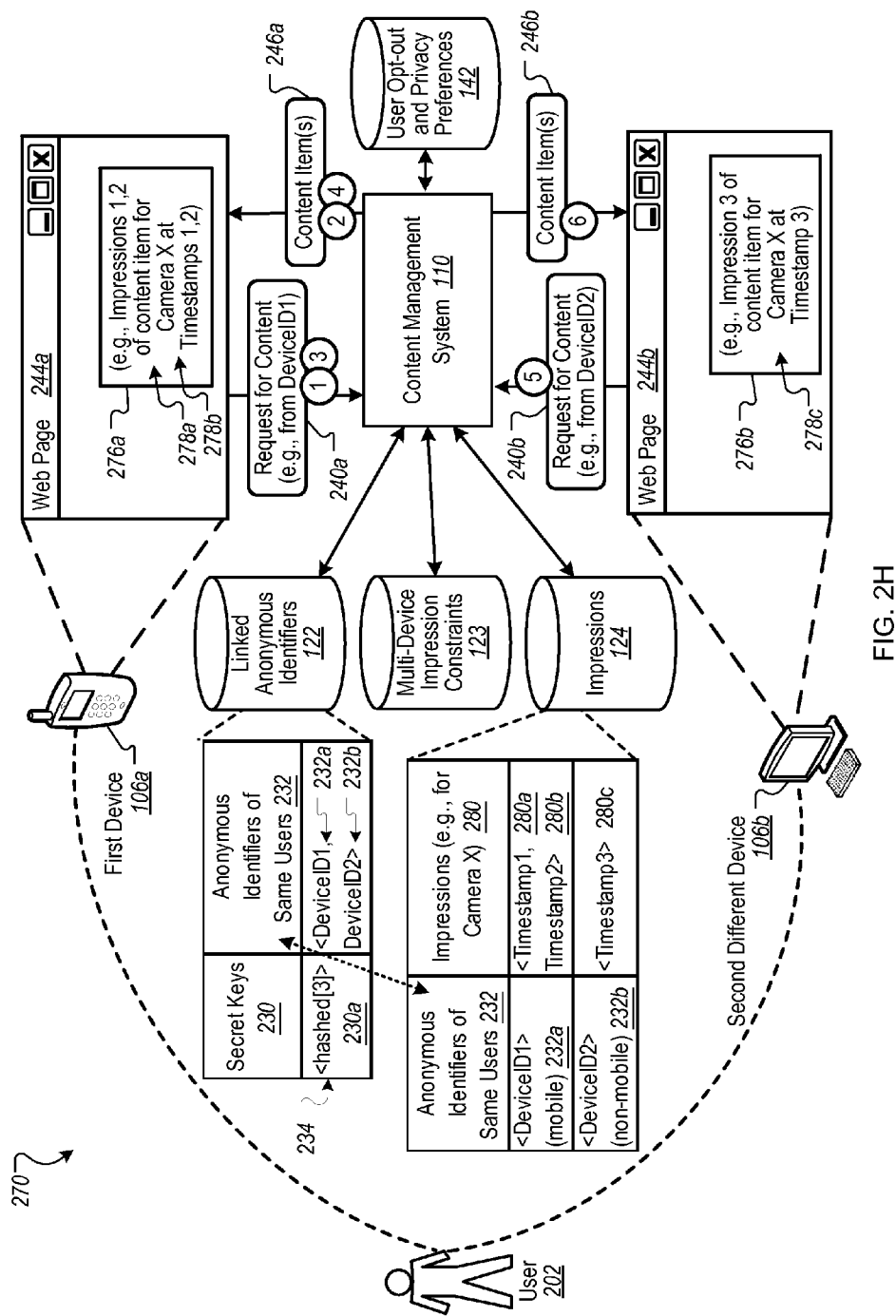


FIG. 2H

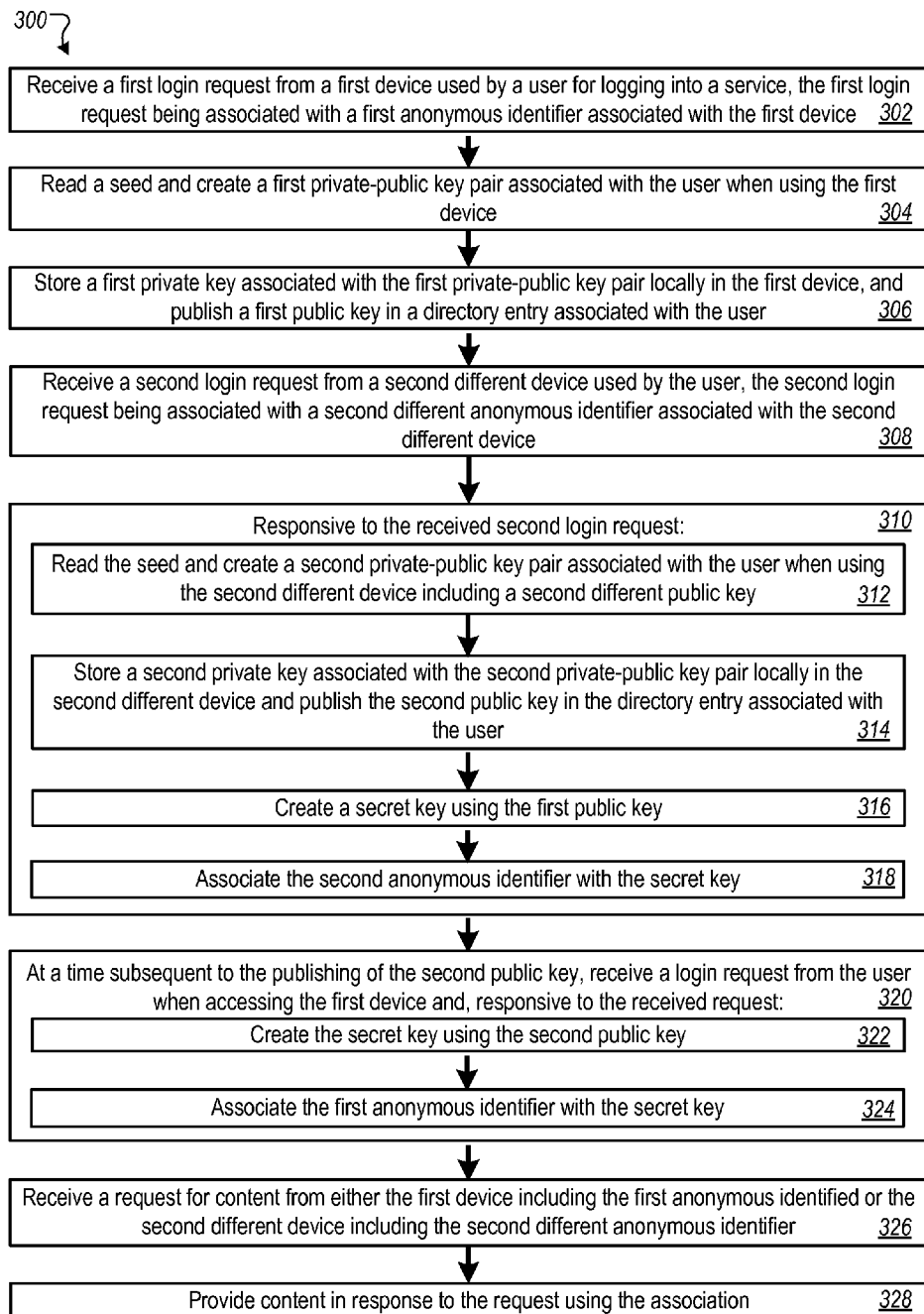


FIG. 3A

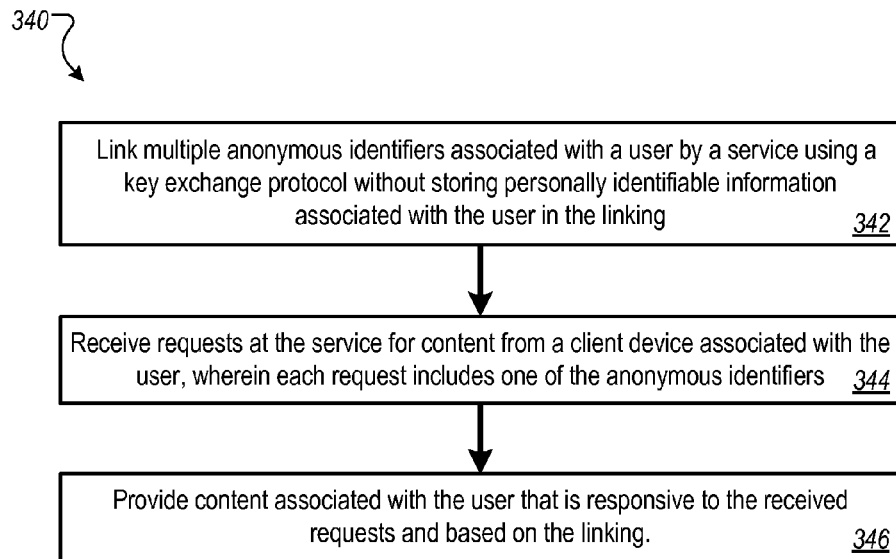


FIG. 3B

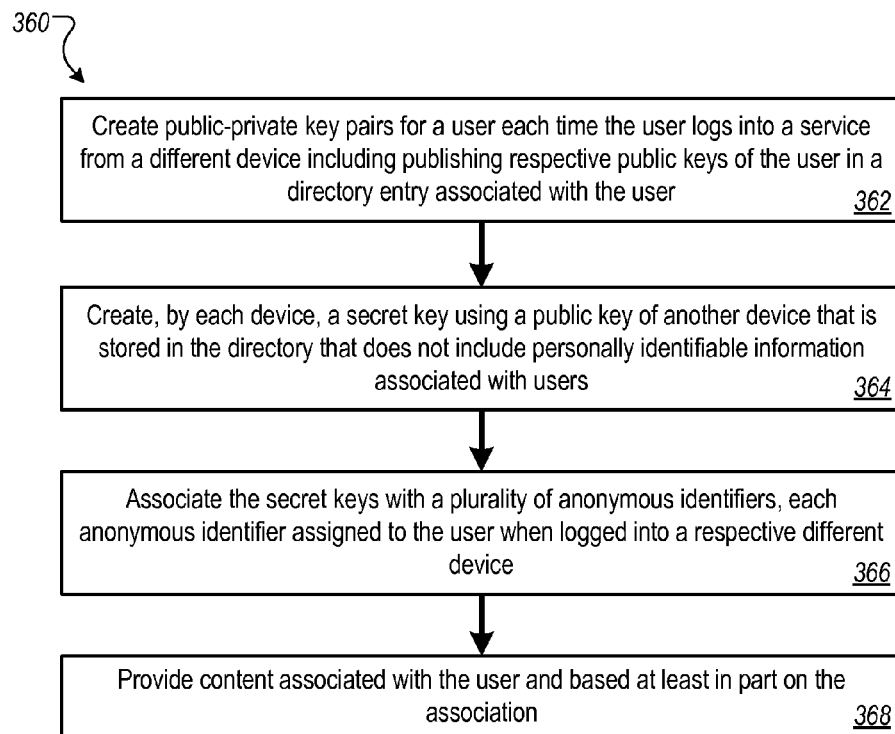


FIG. 3C

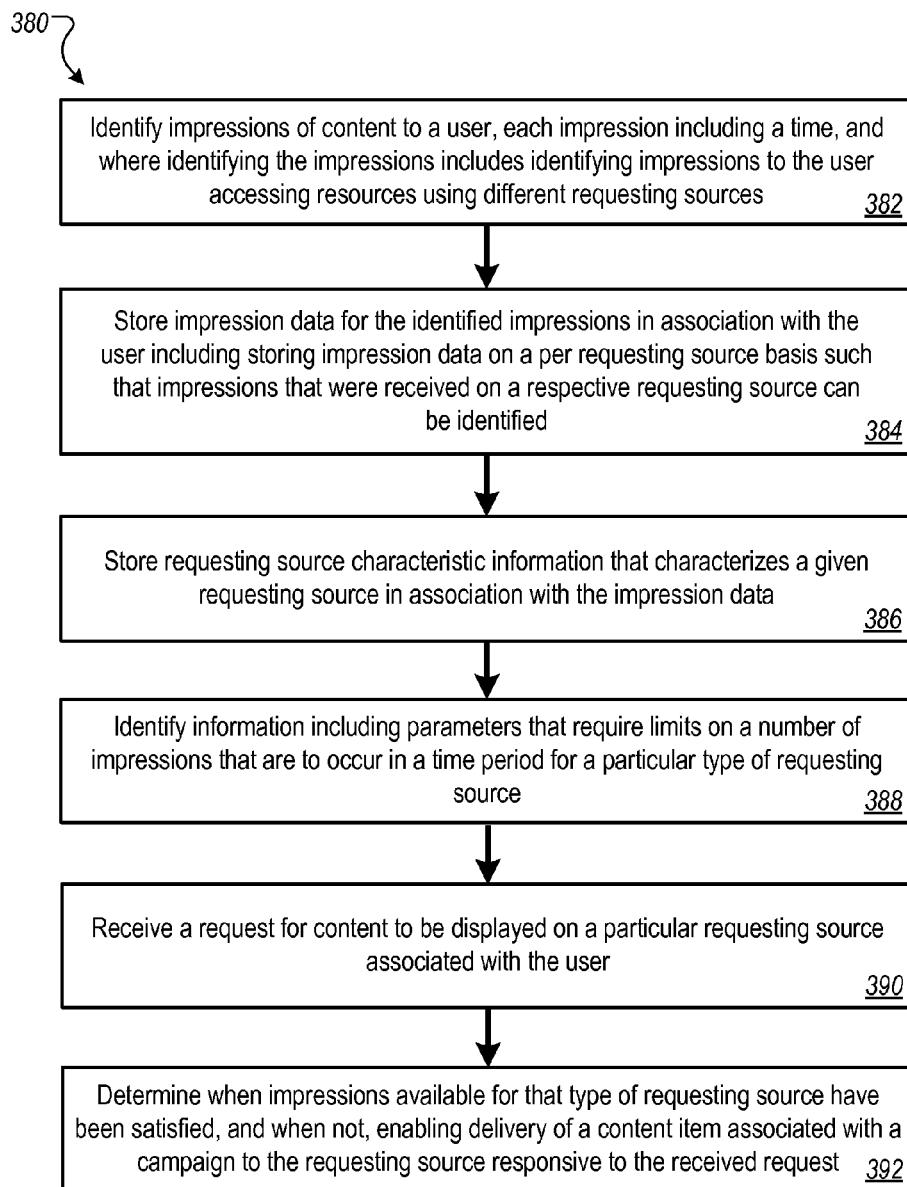


FIG. 3D

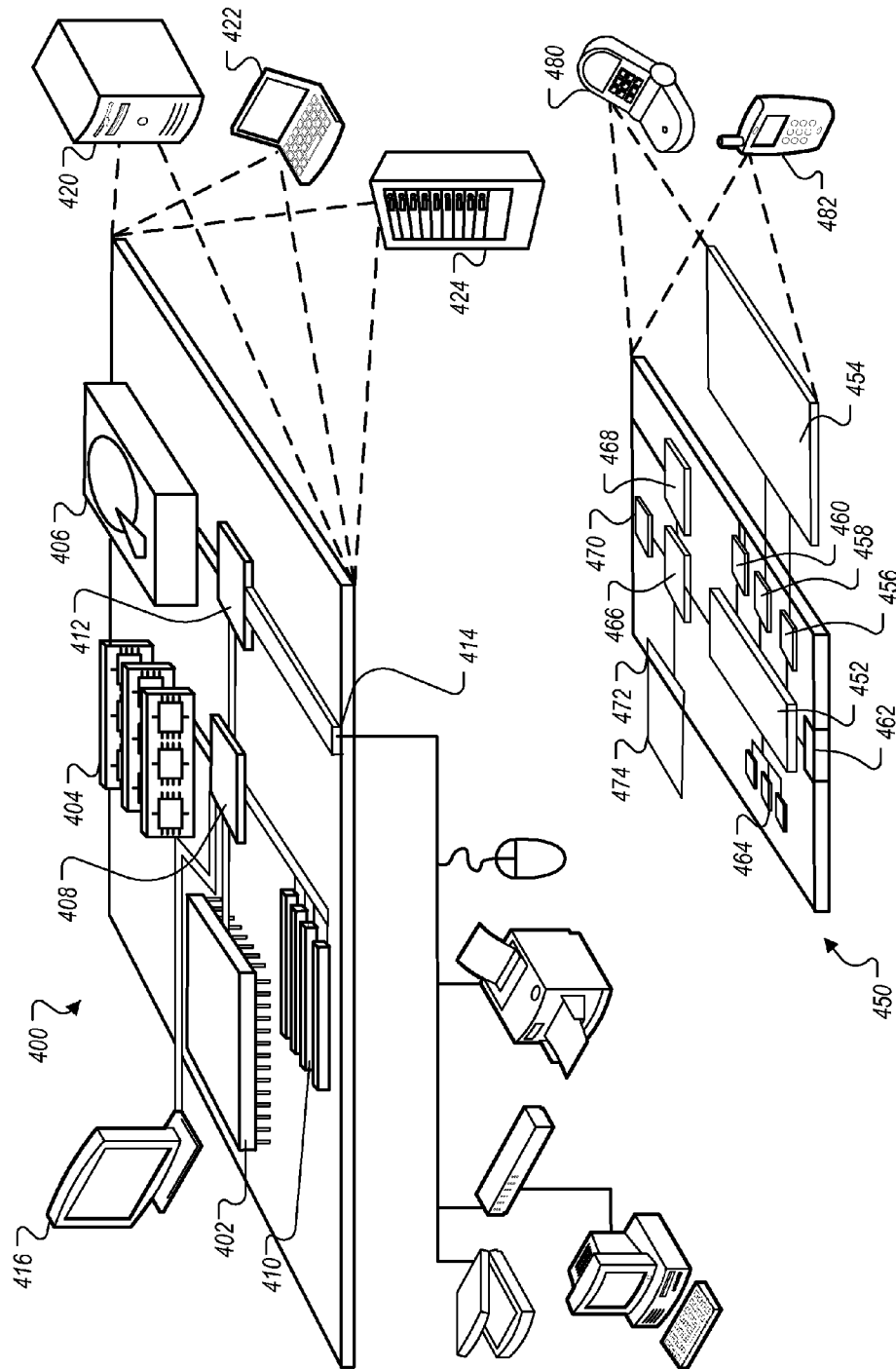


FIG. 4

1

FREQUENCY CAPPING OF CONTENT ACROSS MULTIPLE DEVICES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of and claims priority to U.S. application Ser. No. 13/539,123, filed on Jun. 29, 2012, which is a continuation-in-part of and claims priority to U.S. application Ser. No. 13/458,124, filed on Apr. 27, 2012, the entire contents of which are hereby incorporated by reference.

BACKGROUND

This specification relates to information presentation.

The Internet provides access to a wide variety of resources. For example, video and/or audio files, as well as web pages for particular subjects or particular news articles, are accessible over the Internet. Access to these resources presents opportunities for other content (e.g., advertisements) to be provided with the resources. For example, a web page can include slots in which content can be presented. These slots can be defined in the web page or defined for presentation with a web page, for example, along with search results.

Content item slots can be allocated to content sponsors as part of a reservation system, or in an auction. For example, content sponsors can provide bids specifying amounts that the sponsors are respectively willing to pay for presentation of their content. In turn, an auction can be run, and the slots can be allocated to sponsors according, among other things, to their bids and/or the relevance of the sponsored content to content presented on a page hosting the slot or a request that is received for the sponsored content. The content can then be provided to the user on any devices associated with the user such as a personal computer (PC), a smartphone, a laptop computer, or some other user device.

SUMMARY

In general, one innovative aspect of the subject matter described in this specification can be implemented in methods that include a computer-implemented method for delivering content. The method comprises identifying impressions of content to a user, each impression including a time, wherein identifying impressions includes identifying impressions to the user accessing resources using different requesting sources. The method further comprises storing impression data for the identified impressions in association with the user including storing impression data on a per requesting source basis such that impressions that were received on a respective requesting source can be identified. The method further comprises storing requesting source characteristic information that characterizes a given requesting source in association with the impression data. The method further comprises identifying information including parameters that require limits on a number of impressions that are to occur in a time period for a particular type of requesting source. The method further comprises receiving a request for content to be displayed on a particular requesting source associated with the user. The method further comprises determining when impressions available for that type of requesting source have been satisfied based at least in part on the impression data and the parameters, and when not, enabling delivery of a content item associated with a campaign to the requesting source responsive to the received request.

2

These and other implementations can each optionally include one or more of the following features. The content item can be an advertisement. Identifying impressions can include counting a number of impressions of the content item that have been made on the various requesting sources. The information can include threshold limits on a per requesting source basis for impressions of the content item. The requesting sources can include different user devices, different browsers or different applications. Storing impression data can include storing impression data in association with a cookie that is linked to a requesting source. Cookies of requesting sources associated with a user can be linked using a Diffie-Hellman key protocol. The cookies can be linked using a secret key derived from a seed that is unique to the user. The method can further comprise storing a mapping of cookies associated with the user including storing the impression data for each cookie. The requesting source characteristic information can characterize a type of device, and the type of device can be selected from a group comprising a mobile device, a desktop device, a tablet or other device. The information can include limits for two different requesting source types. Receiving a request can include receiving a request for an advertisement to fill a content slot on a resource. Determining impressions available can include comparing a number of impressions in the impression data for a given device type associated with the received request with a threshold number of impressions for the given device type as specified in the information for the campaign. The method can further comprise receiving the information from a content sponsor associated with the campaign. The method can further comprise determining the information to enable satisfaction of one or more goals of a serving system that serves the content item in response to the received request. The information further can include parameters that require limits on a total number of impressions that are to occur in a time period across a set of the different requesting sources, and the method can further comprise determining when impressions available for the set of the different requesting sources have been satisfied, and when not, enabling delivery of the content item associated with the campaign to the requesting source within the set of the different requesting sources and responsive to the received request. The information can include sequencing information for sequencing content items in the campaign, and identifying impressions and storing impression data can further include identifying a sequence of impressions of content items from the campaign. Determining when impressions are available can include determining a next content item in a sequence to deliver responsive to the request based at least in part on the stored impression data.

Particular implementations may realize none, one or more of the following advantages. Content can be provided to a user based at least in part on prior delivered content, such as content previously delivered to a user on one of a plurality of different devices. Associations among anonymous identifiers can be used enable delivery of interesting content to a user. Content sponsors can be provided with more precise mechanisms for delivering content to users.

The details of one or more implementations of the subject matter described in this specification are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of an example environment for delivering content.

FIGS. 2A through 2E collectively show an example system for providing content to a user who is recognized when using multiple different devices.

3

FIG. 2F shows example calculations of public, private and secret keys.

FIG. 2G shows an example content sponsor interface for defining frequency capping settings across multiple requesting sources.

FIG. 2H is a block diagram that depicts an example sequence of events in a system for frequency capping of impressions of content items to multiple devices associated with a same user.

FIG. 3A is a flowchart of an example process for providing content to a user on any of multiple devices associated with the user.

FIG. 3B is a flowchart of an example process for providing content to a user on any of multiple devices associated with the user.

FIG. 3C is a flowchart of an example process for providing content to a user on any of multiple devices using public-private keys.

FIG. 3D is a flowchart of an example process for limiting impressions of content based on frequency capping across multiple requesting sources.

FIG. 4 is a block diagram of an example computer system that can be used to implement the methods, systems and processes described in this disclosure.

Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

This document describes methods, processes and systems for providing content, including providing frequency capping for the content, to a user having or being associated with multiple devices, without storing personally identifiable information associated with the user. For example, when a user logs onto a user service from a first device (e.g., the user's home PC), a public key-private key pair can be determined and the public key can be published. The public key can be associated with the user's first device and stored by the user service. The private key can be stored locally. Subsequently, when the user logs into the service from a second different device (e.g., a different physical device, browser or application), the second different device can also determine a public-private key pair. Each device can subsequently compute a secret key using the device's own private key and the other device's published public key. The secret key can be stored in combination with anonymous identifiers (e.g., an anonymous cookie) of each device, thus creating a linking or association between the devices. While different cookies are typically associated with different devices, a set of different cookies associated with any of a number of different types of requesting sources used by a user (such as browsers, applications (e.g., games, mobile apps), physical devices (e.g., desktop devices, mobile devices, tablets, smart phones or other physical devices) or any requesting source that requests and receives content) can be linked as will be discussed in further detail below.

For example, a content sponsor can provide frequency capping constraints that indicate that a particular content item (e.g., an advertisement) is to be presented to a user no more than three times in a given time period, regardless of the device on which the content item is presented. In some implementations, content sponsors can specify limits on the number of impressions to provide on a per user device basis, e.g., depending on characteristics of the devices, such as whether the device is mobile versus non-mobile. Content sponsors can also specify sequences of impressions so that, for example, the first two impressions are to be presented on any of the

4

user's mobile devices, and the third impression is to be presented on any of the user's non-mobile devices. While user devices are one example of requesting sources for which content is provided and for which linking can occur, other requesting sources can include different browsers, different applications, or other requesting sources that use identifiers.

In some implementations, the anonymous identifiers can be cookies, browser cookies, device identifiers, or other identifiers that are associated with each device. As a result, the mapping can identify all of the devices associated with the user without storing personally identifiable information (PII) associated with the user. When content is subsequently provided to the user on any of the devices, information included in the mapping can be used to assist in selecting relevant content to be provided to the user. The selection of relevant content can include decisions regarding how content is delivered to the user, such as and including, limitations on when or how content is delivered. For example, the number of impressions of an advertisement can be limited to a fixed number of impressions per user per time period regardless of how many devices the user uses.

In some implementations, anonymous identifiers can be associated with different browsers or other applications on the same device. For example, the techniques described in this disclosure can be used to link two or more identifiers of applications that may have different cookie spaces on the same device, or applications on different devices, or a combination of both.

In some implementations, linking anonymous identifiers can be used in handshaking among mobile applications or a combination of mobile applications, browsers and other applications. For example, mobile applications may each have their own cookie space even on the same device which can prevent handshaking with other applications. Each mobile application can use the techniques described herein to generate, for example, a private key and a public key, publish the public key, access public keys of other mobile applications (or associated with other devices), and compute secret keys using their own private keys and the public keys of other mobile applications (or associated with other devices).

In some implementations, users may be provided with an opportunity to opt in/out of programs or features that allow the user to be discovered across multiple devices and/or to be provided content based on the discovery.

In some implementations, the mapping process can be repeated periodically to ensure that the anonymous identifiers (e.g., cookies) are not stale, thus keeping session history information for the user up-to-date. For example, cookies on a computer can expire over time, or a user can clear a cookie, resulting in setting a new cookie. Repeating the mapping process periodically can ensure that the current set of identifiers belonging to the user are correctly mapped. While reference is made to cookies, other forms of anonymous identifiers can be used including those that include or have been derived from a seed.

In some implementations, user session history information can be stored anonymously. For example, the session history information can include a user's browsing history, the times that the user has seen a particular advertisement, and other session history information. The information can be stored in association with the anonymous identifiers described herein. In some implementations, session history information associated with the user's session on a first device can be stored in a table that includes the anonymous identifier associated with the first device. The same table can also be used to store the same user's session history information for the user's session on a second device. In some implementations, a separate or

5

the same table can be used to store associations among the anonymous identifiers. In some implementations, anonymous identifiers, the associations (e.g., linking to the secret key), and the session data all can be stored, for example, without any corresponding personally identifiable information for a given user.

As will be described in further detail below, subsequent to the storage of the association and session history information, a request for content (e.g., an advertisement) can be sent from any of the devices associated with that user (the request including an anonymous identifier associated with a given device). In some implementations, the session history information stored in the tables can be used in determining, for example, advertisements that may be of interest to the user responsive to the received request. The determination can include inferences for the user based on the user's stored session history information. In some implementations, the session history information for the user can be aggregated, e.g., by joining tables using the anonymous identifiers. For example, a request for content can be received, and the request can include an anonymous identifier associated with a user's desktop device. The received anonymous identifier can be used to look up the user's other anonymous identifiers (e.g., for mobile and other devices of the user). The retrieved set of anonymous identifiers can be used access to session history information in the other tables (e.g., user browsing history). In some implementations, all of the session history information can be joined together for the respective devices producing aggregated information. In some implementations, the aggregated session history information can be provided to a content management system in order to determine and select eligible content for delivery to the user responsive to the received request. For example, because the session history information can include the number of times that the user has seen a particular advertisement, the content management system can help to avoid selecting an advertisement for the user which has already been presented a predetermined number of times.

In some implementations, aggregating the information can occur on demand, e.g., in real time after a request for content occurs. For example, the user's session history information, stored individually by anonymous identifier in the various tables, can be joined. Aggregating the information in real time can solve issues, for example, related to whether the user has opted out of being provided content based on the devices used by the user. For example, session history information for a device for which the user has opted out will not be aggregated with other session history information. In some implementations, the information for a user can be aggregated and stored in advance of any requests for content. For example, all of the user session history information can be stored in a third table, e.g., that includes all of the user session history information across all of the user's devices.

FIG. 1 is a block diagram of an example environment **100** for delivering content. The example environment **100** includes a content management system **110** for selecting and providing content in response to requests for content. The example environment **100** includes a network **102**, such as a local area network (LAN), a wide area network (WAN), the Internet, or a combination thereof. The network **102** connects websites **104**, user devices **106**, content sponsors **108** (e.g., advertisers), publishers **109**, and the content management system **110**. The example environment **100** may include many thousands of websites **104**, user devices **106**, content sponsors **108** and publishers **109**.

In some implementations, the example environment **100** further includes a user login service **120** that can provide, for

6

any particular user, access to the user's Web services, e-mail, social networks, business applications or other resources. For example, the user login service **120** can receive login requests from the user, such as through a Web browser or other application running on any device associated with the user. The login request can include, for example, the user's login ID (e.g., a unique identifier, an email address, a phone number, or any other identifier for the user that can be used for verifying the user at login). The user login service **120** can also maintain information related to the devices on which the user is currently logged on, or has been logged into recently. The information can include, for example, a mapping of anonymous identifiers for the devices with a session key that does not contain personally identifiable information associated with the user. In some implementations, the mapping can be stored, for each user, in a data store of linked anonymous identifiers **122**, or in some data structure.

In some implementations, the user login information **121** or some other data store can store user login IDs, public keys and initial seeds. For example, as described below a user's first device can access the public key published by a user's second device. At the same time, seed values can be read from the user login information **121** by any of the user's devices and used to determine a secret key.

The environment **100** can include plural data stores by which information can be stored that is associated with frequency capping across multiple requesting sources of a user. For example, a data store of multi-device impression constraints **123** can store information provided by content sponsors that identifies frequency capping constraints for impressions of content items to users having multiple user devices or other requesting sources. The information can include limits of the number of impressions that are provided in total or based on device type (e.g., mobile versus non-mobile, Browser A versus Browser B, etc.) or other characteristics, or based on sequences of impressions (e.g., two impressions on mobile devices followed by one impression on a non-mobile device). Other frequency capping information is possible. An impressions database **124** can be used to track, for each content item, the requesting sources on which each impression has occurred and an associated timestamp indicating a date and time of the impression. For example, the content management system **110** can update the impressions database **124** as impressions of content items occur and use the information, e.g., impressions already provided in a particular time period, to determine whether an impression is to be provided to a user device, based on content sponsor-provided frequency capping constraints.

A data store of user opt-out and privacy preferences **142** can include information that the user has provided regarding if and how information about the user's different devices can be used. For example, users can use one or more user preferences web page that may be part of (or separate from) the user login service **120**. In some implementations, users can set a preference that says, "Do not link my different devices," or selectively identify which devices are allowed (or not allowed) to be linked. Then, before any operation is performed that may link the anonymous identifiers of the user's different devices, the user's user opt-out and privacy preferences **142** can be checked, and the linking will be performed only if allowed by the user. In some implementations, the user may specify settings that prohibit providing content based on the linking. For example, while the user may allow his smart phone and PC to be linked, the user may decide that no content (e.g., advertisements) should be provided based on the linking.

A website **104** includes one or more resources **105** associated with a domain name and hosted by one or more servers. An example website is a collection of web pages formatted in hypertext markup language (HTML) that can contain text, images, multimedia content, and programming elements, such as scripts. Each website **104** can be maintained by a content publisher, which is an entity that controls, manages and/or owns the website **104**.

A resource **105** can be any data that can be provided over the network **102**. A resource **105** can be identified by a resource address that is associated with the resource **105**. Resources include HTML pages, word processing documents, portable document format (PDF) documents, images, video, and news feed sources, to name only a few. The resources can include content, such as words, phrases, images, video and sounds, that may include embedded information (such as meta-information hyperlinks) and/or embedded instructions (such as JavaScript scripts).

A user device **106** is an electronic device that is under control of a user and is capable of requesting and receiving resources over the network **102**. Example user devices **106** include personal computers (PCs), televisions with one or more processors embedded therein or coupled thereto, set-top boxes, mobile communication devices (e.g., smartphones), tablet computers and other devices that can send and receive data over the network **102**. A user device **106** typically includes one or more user applications, such as a web browser, to facilitate the sending and receiving of data over the network **102**.

A user device **106** can request resources **105** from a website **104**. In turn, data representing the resource **105** can be provided to the user device **106** for presentation by the user device **106**. The data representing the resource **105** can also include data specifying a portion of the resource or a portion of a user display, such as a presentation location of a pop-up window or a slot of a third-party content site or web page, in which content can be presented. These specified portions of the resource or user display are referred to as slots (e.g., ad slots).

To facilitate searching of these resources, the environment **100** can include a search system **112** that identifies the resources by crawling and indexing the resources provided by the content publishers on the websites **104**. Data about the resources can be indexed based on the resource to which the data corresponds. The indexed and, optionally, cached copies of the resources can be stored in an indexed cache **114**.

User devices **106** can submit search queries **116** to the search system **112** over the network **102**. In response, the search system **112** accesses the indexed cache **114** to identify resources that are relevant to the search query **116**. The search system **112** identifies the resources in the form of search results **118** and returns the search results **118** to the user devices **106** in search results pages. A search result **118** can be data generated by the search system **112** that identifies a resource that is responsive to a particular search query, and includes a link to the resource. In some implementations, the search results **118** include the content itself, such as a map, or an answer, such as in response to a query for a store's products, phone number, address or hours of operation. In some implementations, the content management system **110** can generate search results **118** using information (e.g., identified resources) received from the search system **112**. An example search result **118** can include a web page title, a snippet of text or a portion of an image extracted from the web page, and the URL of the web page. Search results pages can also include one or more slots in which other content items (e.g., ads) can be presented. In some implementations, slots on search

results pages or other web pages can include content slots for content items that have been provided as part of a reservation process. In a reservation process, a publisher and a content item sponsor enter into an agreement where the publisher agrees to publish a given content item (or campaign) in accordance with a schedule (e.g., provide 1000 impressions by date X) or other publication criteria. In some implementations, content items that are selected to fill the requests for content slots can be selected based, at least in part, on priorities associated with a reservation process (e.g., based on urgency to fulfill a reservation).

When a resource **105**, search results **118** and/or other content are requested by a user device **106**, the content management system **110** receives a request for content. The request for content can include characteristics of the slots that are defined for the requested resource or search results page, and can be provided to the content management system **110**.

For example, a reference (e.g., URL) to the resource for which the slot is defined, a size of the slot, and/or media types that are available for presentation in the slot can be provided to the content management system **110**. Similarly, keywords associated with a requested resource ("resource keywords") or a search query **116** for which search results are requested can also be provided to the content management system **110** to facilitate identification of content that is relevant to the resource or search query **116**.

Based at least in part on data included in the request, the content management system **110** can select content that is eligible to be provided in response to the request ("eligible content items"). For example, eligible content items can include eligible ads having characteristics matching the characteristics of ad slots and that are identified as relevant to specified resource keywords or search queries **116**. In some implementations, the selection of the eligible content items can further depend on user signals, such as demographic signals and behavioral signals. Other information, such as user identifier information that is associated with the mappings described above, can be used and/or evaluated when selecting eligible content.

The content management system **110** can select from the eligible content items that are to be provided for presentation in slots of a resource or search results page based at least in part on results of an auction (or by some other selection process). For example, for the eligible content items, the content management system **110** can receive offers from content sponsors **108** and allocate the slots, based at least in part on the received offers (e.g., based on the highest bidders at the conclusion of the auction or based on other criteria, such as those related to satisfying open reservations). The offers represent the amounts that the content sponsors are willing to pay for presentation (or selection) of their content with a resource or search results page. For example, an offer can specify an amount that a content sponsor is willing to pay for each 1000 impressions (i.e., presentations) of the content item, referred to as a CPM bid. Alternatively, the offer can specify an amount that the content sponsor is willing to pay (e.g., a cost per engagement) for a selection (i.e., a click-through) of the content item or a conversion following selection of the content item. For example, the selected content item can be determined based on the offers alone, or based on the offers of each content sponsor being multiplied by one or more factors, such as quality scores derived from content performance, landing page scores, and/or other factors.

A conversion can be said to occur when a user performs a particular transaction or action related to a content item provided with a resource or search results page. What constitutes a conversion may vary from case-to-case and can be deter-

mined in a variety of ways. For example, a conversion may occur when a user clicks on a content item (e.g., an ad), is referred to a web page, and consummates a purchase there before leaving that web page. A conversion can also be defined by a content provider to be any measurable or observable user action, such as downloading a white paper, navigating to at least a given depth of a website, viewing at least a certain number of web pages, spending at least a predetermined amount of time on a web site or web page, registering on a website, experiencing media, or performing a social action regarding a content item (e.g., an ad), such as republishing or sharing the content item. Other actions that constitute a conversion can also be used.

In some implementations, the likelihood that a conversion will occur can be improved, such as by recognizing a user when the user has accessed resources using multiple devices. For example, if it is known that a content item (e.g., an advertisement) has already been seen by a user on a first device (e.g., the user's home PC), then a determination can be made (e.g., through parameters) whether or not to provide the same content item to the same user on a different device (e.g., the user's smartphone). This can increase the likelihood of a conversion, for example, by either repeating impressions of an advertisement or avoiding subsequent impressions, depending on how multiple impressions for the advertisement to the same user are predicted to lead to a conversion in either case.

For situations in which the systems discussed here collect personal information about users, the users may be provided with an opportunity to opt in/out of programs or features that may collect personal information (e.g., information about a user's social network, social actions or activities, a user's preferences or a user's current location). In addition, certain data may be anonymized in one or more ways before it is stored or used, so that personally identifiable information associated with the user is removed. For example, a user's identity may be anonymized so that the no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined.

FIGS. 2A-2E collectively show an example system 200 for providing content to a user who is recognized when using multiple different devices. In some implementations, recognition of the user across different devices can be achieved by linking anonymous identifiers of the user's multiple different devices. As an example, an anonymous identifier 206a of a first device 106a (e.g., a desktop computer of a user 202) can be linked to an anonymous identifier 206b of a second different device 106b (e.g., a laptop computer of the user 202). In some implementations, the system 200 can be part of the environment 100 that is described above with reference to FIG. 1. An example sequence of events (e.g., with numbered steps 0 and 1a through 8) follows for associating the anonymous identifiers 206a and 206b and providing content based on the association. However, other sequences can also be used to link devices 106a, 106b and additional devices 106 associated with the user 202. In some implementations, the devices 106a, 106b and additional devices 106 can be linked using associations stored in the linked anonymous identifiers 122. The associations can be stored, for example, without storing any personally identifiable information for the user 202.

Before any linking occurs using the anonymous identifiers associated with a user's different devices, the user login service 120 (or the content management system 110) can check 107 the user's user opt-out and privacy preferences 142 to see

if the user has opted out of such linking. For example, if the user has specified not to allow the user's devices to be linked (or use information thereof), then steps 2a through 6b will not occur, and the content provided in step 8 may be different.

In some implementations, a first step 1a (e.g., as allowed by the user) of the sequence of steps can occur, for example, when the user 202 logs into the first device 106a (e.g., the user's desktop computer) using a login service (not shown in FIGS. 2A-2D). For example, the login service or some other component can receive a login request 208a from the first device 106a. The login request 208a can be associated with the anonymous identifier 206a (e.g., a cookie or device identifier) associated with the first device 106a. In some implementations, the login request 208a and/or other login requests can be requests to log into a social service.

In some implementations, the user login information 121 can store user login IDs 210, seeds 212 and public keys 214 associated with multiple users. The user login information 121, for example, can serve as a directory that includes one or more entries, each entry indexed by an identifier associated with a given user (e.g., user login identifier, email address, or some other identifier). For example, when the user 202 logs into the device 106a using the login service, information stored for the user in the user login information 121 can include a login ID 210a, a seed 212a (e.g., a generator-prime pair, such as 7, 11, that is usable by all of the user's devices), and, as will be discussed in further detail below, a public key 214. At the current stage of the sequence of steps, the public key 214 has not yet been determined for the current user. In some implementations, seeds 212 can vary by user, e.g., the seed 212b (e.g., generator-prime pair 7, 13) for a second user can be different from the seed 212a.

At step 2a, the first device 106a can read a seed 216a (e.g., a generator-prime pair 7, 11 from the user login information 121) and create a private-public key pair that is associated with the user 202 using the first device 106a. In some implementations, creating the private-public key pair can include, at step 3a, computing 218a a private key (e.g., 9) and computing a public key (e.g., 4). In some implementations, generation of public and private keys can use generator G, prime P pair (e.g., 7, 11), where $G < P$, an example of which is described with reference to FIG. 2F. At step 4a, the public key is published 220a, e.g., stored as the public key 214a. The private key n and the public key 4 constitute the private-public key pair, yet each typically is stored in a different location. For example, the private key n can be stored locally on the first device 106a, e.g., in local storage 219. The public key (e.g., 4) can be stored in the user login information 121 as the public key 214a. In some implementations, the public key 214a can be stored in a row 222 that includes user login information for the user 202 on one or more devices (e.g., devices 106a and 106b in the current example). For example, the row 222 can serve as a directory entry associated with the user 202. Each of the other rows can be used to store information for a different user.

Referring now to FIG. 2B, at step 2b, after a login request (step 1b) by the user on a second different device 106b, a seed 216b (e.g., the generator-prime pair 7, 11) can be read (e.g., from the user login information 121) and a second private-public key pair that is associated with the user can be created. The second private-public key pair is associated with the user 202 using the second device 106b. For example, the second private-public key pair is different than the private-public key pair that is associated with the login by the user 202 on the first device 106a. In some implementations, creating the second private-public key pair can include, at step 3b, computing 218b a private key (e.g., m) and computing a second public

11

key (e.g., 8). At step **4b**, the second public key is published **220b**, e.g., by adding the second public key to the set of public keys stored as the public keys **214a**. The private key **m** and the public key **8** constitute the second private-public key pair (e.g., $\langle m, 8 \rangle$), the values of which are different from those of the private-public key pair computed for the first device **106a** (e.g., $\langle n, 4 \rangle$). In some implementations, the private key **m** can be stored locally on the second different device **106b**, e.g., in local storage **221b**. The public key (e.g., 8) can be stored, for example, in user login information **121** with the public key **4** from the first device **106a**. As a result, the directory entry stored in the row **222** (and associated with the user **202**) is updated, now including two public keys.

Referring now to FIG. 2C, at step **5a**, the second different device **106b** can create a secret key **226a** (e.g., 3) using the public key (e.g., 4) from the first device **106a** and the second private key (e.g., private key **m** from local storage **221b**). At step **6a**, the second different device **106b** can also associate **228a** the second anonymous identifier (e.g., "Device ID 2") with the secret key (e.g., 3). In some implementations, the association can include storing the association, e.g., in the linked anonymous identifiers **122**. For example, the linked anonymous identifiers **122** can include secret keys **230** and anonymous identifiers of the same users **232**. For example, a row **234** can include a secret key **230a** (e.g., 3 or a hashed representation of 3) and the anonymous identifier **232b** (e.g., "Device ID 2") that corresponds to the association that occurred in step **6a**. At a time subsequent to the publishing of the second public key (e.g., 8), and after the secret key 3 has been computed and an association stored (e.g., as a hashed representation) with the second different device **106b**, the user may log in again at the first device **106a**. As a result, a login request **208c** can be received (e.g., by the login service) from the user at the first device **106a**. For example, the login request **208c** can be similar to the login request **208a** described above. However, in this case, the login service, for example, can determine that a public key exists for another device associated with the user, e.g., the second different device **106b**. Using the additional public key, a link or association can be made between the two devices **106a** and **106b** as described in further detail below. In some implementations, whenever the secret key is stored, the stored value can be a hashed version of the secret key, e.g., using a one-way hash function.

Referring now to FIG. 2D, at step **5b**, in response to the login request **208c**, the first device **106a** can create a secret key **226b** (e.g., 3) using the public key (e.g., 8) from the second different device **106b** and the first private key (e.g., private key **n** from local storage **221a**). For example, the secret key can match the secret key computed by the second different device **106b**. At step **6b**, the first device **106a** can also associate **228b** the second anonymous identifier (e.g., Device ID 2) with the secret key (e.g., 3). In some implementations, the association can include storing the association, e.g., in the linked anonymous identifiers **122**. For example, the row **234** containing the secret key **230a** (e.g., 3) and the anonymous identifier **232b** (e.g., "Device ID 2") can be updated to also include the anonymous identifier **232a** (e.g., "Device ID 1"). As a result of storing the association, the anonymous identifiers **206a** and **206b**, as well as the devices **106a** and **106b**, are now linked. Further, the association among the user's various devices is achieved without storing any personally identifiable information associated with the user.

In some implementations, it is possible that one or more anonymous identifiers such as anonymous identifier **232a** or anonymous identifier **232b** can appear in multiple rows (e.g.,

12

three or more) in the linked anonymous identifiers **122**. This can be an indication, for example, that the device associated with the anonymous identifier is a shared device (e.g., at a library or an Internet café). In this example, the logins by several different users (e.g., three or more) would result in the creation of multiple rows in the anonymous identifiers **122**, each having the same anonymous identifier. In some implementations, when highly-shared devices are detected in this way, the highly-shared devices can be un-linked, or other considerations can be taken. For example, thresholds can be established, and if a cookie or other anonymous identifier appears in more than three rows, the associated can be considered a shared machine.

Referring to FIG. 2E, at step **7**, the content management system **110** can receive a request for content **240a** or **240b** (e.g., a request for advertising content) from either the first device **106a** or the second different device **106b**. For example, the request for content **240a** can be a request for an advertisement to fill an advertisement slot **242a** on a web page **244a** displayed on the first device **106a**. In another example, the request for content **240b** can be a request for an advertisement to fill an advertisement slot **242b** on a web page **244b** displayed on the second different device **106b**. If the request for content **240a** is from the first device **106a**, for example, then the request for content can include the first anonymous identifier **232a**. Otherwise, if the request for content **240b** is from the second different device **106b**, for example, then the request for content can include the second different anonymous identifier **232b**.

Regardless of where the request for content originates, at step **8**, the content management system **110** can provide a content item (e.g., content items **246a** or **246b**) in response to the request and using the association that maps the user **202** to multiple devices (e.g., from the linked anonymous identifiers **122**). For example, the association can be represented by information in the row **234** that associates anonymous identifiers **232a** and **232b**, e.g., based on the same secret key **230a**. Using this information, the content management system **110** can, for example, treat the requests for content as if they originate from the same user, regardless of the particular user device. In some implementations, identifying eligible content items for the request for content **240b**, for example, can depend on content already provided to the same user **202** on the first device **106a**. As a result, an advertisement for California vacations, for example, that is intended for one impression per user can be shown on the first device **106a** and not repeated again on the second different device **106b**. In some implementations, it can be beneficial to provide the same advertisement once and only once to each of the user's multiple devices.

Devices **106a** and **106b** are two examples of devices that the user **202** may use. For example, the user **202** may use a third different device **106c** (e.g., a smart phone). When the user **202** uses the third different device **106c** to log in, for example, the user login service **120** can store a third different anonymous identifier **232** in the linked anonymous identifiers **122**. As a result, all three devices **106a-106c** can be associated with the user **202**, e.g., using the secret key **230a**.

Similarly, other users can use the user login service **120** for logging in from multiple different devices. As a result of a second user logging into a fourth and a fifth device **106**, for example, the user login service **120** can store fourth and fifth different anonymous identifiers in the linked anonymous identifiers **122** (e.g., stored in association with the second user using a secret key **230** that is different from the secret key **230a**).

13

FIG. 2F shows example calculations of public, private and secret keys. Device A calculations 250 provide examples for computing a public key, a private key and a secret key on a first device, e.g., the first device 106a. Device B calculations 252 provide examples for computing a public key, a private key and a secret key on a second different device, e.g., the second different device 106b. Other methods can be used to determine public, private and secret keys.

In some implementations, the calculations can occur in steps, e.g., steps 254a-254e. For example, in step 1 254a, both devices A and B can exchange a prime P (e.g., 11) and a generator G (e.g., 7). In some implementations, the prime and generator can be stored in the user login information 121, as described above. For example, a prime and a generator that is unique to a user (and the devices associated with the user) can be determined and stored at a time that one or more entries in the user login information 121 are created and stored.

In step 2 254b, each device can generate its own private key, e.g., using a random number or in some other way. For example, device A's private key can be 6, and device B's private key can be 9. These private keys can be used in combination with at least the generator and prime from step 1 254a to determine public and private keys in the following steps.

In step 3 254c, each device can compute a public key. In some implementations, computing the public key can use a formula that includes the generator raised to the power of the device's private key, and a modulo P can be performed on the result. Using the generator, prime, and each of the devices' private keys, the resulting public keys for the devices can result in being 4 and 8, respectively.

At step 4 254d, once the public keys are determined, the devices can share their public keys, e.g., by publishing the keys in the user login information 121 as described above. As a result, device A can know device B's public key (e.g., 8), and device B can know device A's public key (e.g., 4).

At step 5 254e, secret keys can be computed, e.g., using a formula that raises the other device's public key to power of the current device's private key, and the result can undergo a modulo P (prime). As a result of the calculations, the secret key for the first and second devices can be 3. Once the secret key is determined, the value can be used by either device to update the row in the linked anonymous identifiers 122 with the device's anonymous identifier. This can be repeated for any other device associated with the same user that computes the secret key using its own private key and the public key from one of the other devices.

FIG. 2G shows an example content sponsor interface 256 for defining frequency capping settings 258 across multiple requesting sources. For example, the frequency capping settings 258 can be one of several data input areas in the content sponsor interface 256 that can be used by a content sponsor to define parameters associated with a campaign, including parameters for providing impressions of advertisement creatives (e.g., creative 259 for Camera X) or other content. The frequency capping settings 258, for example, can be used in combination with information about linked devices or other requesting sources associated with a user. For example, the linking can be accomplished using a Diffie-Hellman key exchange protocol, as described herein, or the linking can be done in some other way. In particular, frequency capping, as described in this disclosure, can make use of cookies, user devices or requesting sources that are linked in some way.

In some implementations, the frequency capping settings 258 can include a number of impressions area 260 in which the content sponsor can specify various maximum numbers of impressions to be provided in total and/or on a per requesting

14

source basis. For example, the number of impressions area 260 can include separate controls for specifying different numbers of impressions, e.g., using a total impressions control 262a, a mobile impressions control 262b, and a non-mobile impressions control 262c. Other controls are possible, e.g., to specify the number of impressions for particular browsers, particular games, or specific (or categories of) other requesting sources. In some implementations, other controls can be provided by which the content sponsor can indicate that impressions are to stop under certain circumstances, such as if a conversion is achieved. In some implementations, impressions can stop automatically by default, e.g., if a conversion occurs and/or if other predefined conditions are met.

In the example shown in FIG. 2G, the content sponsor may specify two as a maximum number of impressions in the mobile impressions control 262b, and one as a maximum number of impressions in the non-mobile impressions control 262c. The content sponsor may do this, for example, in light of historical knowledge that a conversion by a user is more likely to occur if the user sees an advertisement twice on a cell phone (or other mobile device) and once on a non-mobile device (e.g., a personal computer at home). The content sponsor may also make these selections for other reasons, such as a preference for having a certain mix of impressions on mobile versus non-mobile devices. Other ways of determining an appropriate number of impressions per requesting source are possible, e.g., by percentages, such as 50% of impressions on each of mobile and non-mobile devices, or two-thirds of the impressions on mobile.

In some implementations, controls for setting numbers of impressions may not become available for use by the content sponsor until, for example, the content sponsor selects a number of impressions control 262 to enable the number of impressions area 260. For example, until the checkbox associated with the number of impressions control 262 is checked by the content sponsor, the controls 262a-262c may be hidden or disabled (e.g., greyed out). As such, setting the number of impressions, e.g., across multiple devices, can be an optional feature for the campaign sponsor.

In some implementations, a time constraints control 266a can be included. For example, selecting the time constraints control 266a can cause the presentation of additional controls for setting a time period for which the frequency capping is to occur. As an example, the content sponsor may specify that three total impressions are to occur over a day, a week, or some other time period T. Other time constraints can be used, including multiple time periods (e.g., no more than three impressions in a single day and ten impressions during a week). Yet other time constraints can be used to specify time-of-day constraints for presenting content.

Based on the example settings depicted in the number of impressions area 260 of two and one, respectively, for the mobile impressions control 262b and the non-mobile impressions control 262c, impressions may occur in any order. For example, the single impression provided on the user's non-mobile device may be the first, second or third impression, as long as the total number of impressions across all devices is capped at three.

In some implementations, the frequency capping settings 258 can include a sequence of impressions area 268 in which the content sponsor can specify one or more sequences of impressions to be provided to a user over multiple requesting sources. For example, the sequence of impressions area 268 can include sequence controls 270a-270b. As an example, the content sponsor can specify two impressions to mobile devices using sequence control 270a, followed by one impression on non-mobile devices using sequence control

15

270b. Additional controls, each for another count of impressions in the sequence, can be added using a control 272. Other selections are possible, e.g., to specify sequences that include content sponsor-selected numbers of impressions to particular browsers, particular applications, or other types of requesting sources.

In the example shown in FIG. 2G for the sequence of impressions area 268, the content sponsor may, for example, specify two mobile impressions followed by one non-mobile impression. The content sponsor may do this, for example, because of historical knowledge that a conversion is more likely to occur if a user first sees an advertisement twice on a cell phone (or other mobile device) followed by a single impression on a non-mobile device (e.g., a personal computer at home). In some implementations, the sequencing can be arranged so that the latter impressions occur on a non-mobile device, e.g., that is likely to have a user interface and keyboard capabilities that make it easier for entering purchase, registration or other information, or to allow the user to easily perform additional research.

In some implementations, controls for sequences of impressions may not become available unless the content sponsor selects a sequence of impressions control 274, e.g., enabling the sequence of impressions area 268. For example, until the checkbox associated with the sequence of impressions control 274 is checked by the content sponsor, controls such as the controls 270a-270b and 272 may be hidden or disabled. In this way, setting the sequence of impressions, e.g., across multiple devices, can be an optional feature for the campaign sponsor.

In some implementations, content sponsors can specify which user interactions are to be considered to be conversion events for the purposes of frequency capping. For example, the content sponsor may be able to select individual checkboxes to include, as conversion events, walking in to a store, paying using a mobile wallet within a store, using a barcode scanner to scan a product, making a phone call to the advertiser's business, searching for the product online after seeing an advertisement for the product, following the product within a social network, or other events.

FIG. 2H is a block diagram that depicts an example sequence of events in a system 270 for frequency capping of impressions of content items to multiple devices of the same user. For example, the system 270 can limit impressions of content to one or more of user devices 106a and 106b. Frequency capping can be based on frequency capping constraints defined for content that may be provided to any or a defined set of the user's linked devices. For example, frequency capping can occur in light of any impressions that have already been provided to any of the linked user devices 106a and 106b associated with user 202. Frequency capping can be constrained to a time period, e.g., no more than N impressions, over time period T, to the same user, in light of the multiple linked devices associated with the user.

The linked devices 106a and 106b are two examples of, more generally, requesting sources that can include different user devices, different browsers, different applications and/or other requesting sources. In some implementations, the linking can be accomplished using a Diffie-Hellman key exchange protocol as described herein. Other ways of linking identifiers (and thus requesting sources, devices, etc.) can be used.

An example sequence of events for providing frequency capping for multiple linked devices can start at step 1 when the content management system 110 receives the request for content 240a from the first device 106a. For example, the request for content 240a can be a request for an advertisement

16

to fill a content item slot 276a on the web page 244a displayed on the first device 106a. The request for content can include or identify the first anonymous identifier 232a that is associated with the first device 106a, e.g., a mobile device such as a cell phone.

At step 2, the content management system 110 can provide a content item (e.g., content item 246a) in response to the request and using the association that maps the user 202 to multiple devices (e.g., from the linked anonymous identifiers 122). For example, the content item 246a that is ultimately selected by the content management system 110 may be one of several eligible content items that satisfy the conditions for the request for content 240a. As an example, the selected content item 246a may be a Camera X advertisement for which a content sponsor has provided frequency capping constraints across multiple devices of the same user. If the user has not previously seen the Camera X advertisement, then the Camera X advertisement can be provided as a first impression 278a. Selection of content in this example can be based on characteristic information (e.g., mobile vs. non-mobile) that characterizes each requesting source (e.g., user devices 106a and 106b). For example, the Camera X advertisement can be chosen in part because the first device 106a, a mobile device, satisfies the frequency capping constraints defined by the content sponsor.

At the time that content is selected in response to the request for content, the characteristic information is already stored, e.g., in the linked anonymous identifiers 122. The stored information is then used in accordance with criteria used to select content to provide to users of the requesting source. For example, the Camera X advertisement may be selected within the confines of multi-device frequency capping for various reasons. Specifically, the content sponsor of the Camera X advertisement may have designated that at least one impression is to be provided to a mobile device (e.g., the first device 106a). If, for example, the content sponsor had instead designated that Camera X advertisements are never to be provided to mobile devices, then an advertisement different from the Camera X advertisement would be chosen by the content management system 110.

In another example, the content sponsor may have used sequencing constraints to designate that the first impression is to be provided to a mobile device (e.g., the first device 106a). For example, the Camera X advertisement may be chosen by the content management system 110 because the campaign associated with the advertisement includes constraints for showing the Camera X advertisement either as a first impression to a mobile device, or at least once in a campaign that also shows the advertisement to non-mobile devices.

In a counter example, it is possible that the user 202 may have been using the second different device 106b (e.g., a non-mobile device), and an opportunity to provide the Camera X advertisement may have arisen. However, in this example, the content management system 110, using the content sponsor's multi-device impression constraints 123 (including sequencing constraints), would have not considered the Camera X advertisement because the second different device 106b is not mobile.

When the Camera X advertisement is provided as the first impression 278a, the impression is identified as associated with user 202, along with a timestamp identifying the date and time that the impression occurred. The information, e.g., impression information 280a, can be stored in impressions database 124 and associated with anonymous identifier 232a (e.g., "Device ID 1"), corresponding with the requesting source (e.g., first device 106a). For example, impression information 280a can identify the first of potentially multiple

17

impressions (identified by impression information **280**) for the Camera X advertisement to user **202** having linked devices, the first impression occurring on the first user device **106a**. In general, the impressions database **124** can store impression data on a per requesting source basis such that impressions that were received on a respective requesting source can be identified, e.g., to insure that frequency capping constraints are being met.

In some implementations, prior to serving an otherwise eligible content item, the content management system **110** can first access impression information in impressions database **124**. The content management system **110** can then make a determination regarding whether impressions available for that requesting source have been satisfied, and if not, enable delivery of an eligible content item associated with a campaign to the requesting source responsive to the received request. For example, at step **3**, another request for content **240a** can be received from the same mobile device. If the Camera X advertisement is once again an eligible content item, then the content management system **110** can access impression information **280** about previous impressions of the Camera X advertisement to user **202**. The content management system **110** can determine, for example, whether providing the Camera X advertisement would meet the multi-device impression constraints defined by the content sponsor (e.g., including time period constraint **T**). If so, then the Camera X advertisement can be selected. Thus, at step **4**, the content item **246a** can be provided and can be the same Camera X advertisement that the user has previously seen on the first mobile device **106a**. As this is now a second impression **278b** of the Camera X advertisement, impression information **280b** can be added to the impressions database **124**, the information identifying the impression of the advertisement and a corresponding timestamp. At this point in time, two impressions **278a** and **278b** have been presented on the first user device **106a**, both of which are within the time period **T**. Moreover, because the first user device **106a** is a mobile device, the two impressions can satisfy two constraints of a given sponsor. In one example above, constraints included two mobile impressions with one (yet-to-occur) non-mobile impression. In this example, the second impression **278b** also could have been presented on a different mobile device. When a subsequent request for content is received from the first user device **106a** (or some other mobile device) within the time period **T**, the content management system **110** would not select the Camera X advertisement (i.e., based on the example constraints).

At step **5**, the content management system **110** can receive a request for content **240b**, e.g., during the same time period **T**, but this time from the second user device **106b**, a non-mobile device. The Camera X advertisement may again be an eligible content item that the content management system **110** identifies based on the request for content **240b**. Given that the Camera X advertisement is associated with frequency capping constraints, the content management system **110** can, for example, check the multi-device impression constraints **123** to determine which constraints regarding limits or sequences of impressions are in place. For example, the constraints may allow an impression to a non-mobile device during the same time period **T**, either in a mobile/mobile/non-mobile sequence or in a set of two mobile impressions and one non-mobile impression. The content management system **110** can also check impressions database **124**, e.g., determining from impression information **280a** and **280b** that two mobile impressions **278a** and **278b** have already been provided to the user. Because a third impression **278c** is allowed to occur in light of the first and second impressions **278a** and

18

278b, the content management system **110** can provide the Camera X advertisement as a content item **246b** at step **6**.

During the remainder of time period **T**, no other impressions of the Camera X advertisement would be presentable to user **202**, regardless of which device requested content. However, once the time period **T** expires, a new period including new frequency capping constraints can be used in association with a decision as to whether or not present the advertisement. In some implementations, impression information in the impressions database **124** can be purged when, for example, entries include outdated timestamps (e.g., having an age greater than time period **T**).

FIG. **3A** is a flowchart of an example process **300** for providing content to a user on any of multiple devices associated with the user. In some implementations, the content management system **110** and/or the user login service **120** can perform steps of the process **300** using instructions that are executed by one or more processors. FIGS. **1-2F** are used to provide example structures for performing the steps of the process **300**.

A first login request is received from a first device used by a user for logging into a service, the first login request being associated with a first anonymous identifier associated with the first device (**302**). For example, referring to FIG. **2A**, the user login service **120** can receive the login request **208a** from the first device **106a** (e.g., a personal computer) being used by the user **202**. The login request can be associated, for example, with the anonymous identifier **206a** (e.g., "Device ID 1") that is associated with the first device **106a**.

A seed is read, and a first private-public key pair is created that is associated with the user when using the first device (**304**). As an example, the user login service **120** can read the seed **212a** (e.g., generator-prime pair **7, 11**) and provide the seed **212a** to the first device **106a**. Using the seed, the first device **106a** can determine the private key (e.g., **9**) and the public key (e.g., **4**) associated with first device **106a**.

A first private key associated with the first private-public key pair is stored locally in the first device, and a first public key is published in a directory entry associated with the user (**306**). The first device **106a**, for example, can store the private key in local storage **221a**. The first device **106a** can also provide the public key (e.g., **4**) to the user login service **120** for storage in user login information **121**.

A second login request is received from a second different device used by the user, the second login request being associated with a second different anonymous identifier associated with the second different device (**308**). As an example, referring to FIG. **2B**, the same user **202** can log into the second different device (e.g., a laptop computer). The user login service **120**, for example, can receive the login request **208b**. The login request can be associated, for example, with the anonymous identifier **206b** (e.g., "Device ID 2") that is associated with the second different device **106b**.

Responsive to the received second login request (**310**), the seed is read, and a second private-public key pair is created that is associated with the user when using the second different device including a second different public key (**312**). As an example, the user login service **120** can read the seed **212a** (e.g., generator-prime pair **7, 11**) and provide the seed **212a** to the second different device **106b**. Using the seed, the second different device **106b** can determine its private key (e.g., **6**) and the public key (e.g., **8**).

A second private key associated with the second private-public key pair is stored locally in the second different device, and the second public key is published in the directory entry associated with the user (**314**). The second different device **106b**, for example, can store the private key in local storage

19

221b. The second different device **106b** can also provide the public key (e.g., **8**) to the user login service **120** for storage in user login information **121**.

A secret key is created using the first public key (**316**). For example, referring to FIG. 2C, the second different device **106b** can compute the secret key **230a** (e.g., **3**) using the public key (e.g., **4**) from the first device and the second different device's own private key (e.g., **6**). Device B calculations **502** shown in FIG. 2F provide example steps and formulas for computing the secret key.

The second anonymous identifier is associated with the secret key (**318**). For example, the second different anonymous identifier (e.g., Device ID **2**) can be stored with the secret key (e.g., a hashed version), e.g., in the linked anonymous identifiers **122**, which is stored separately from the user login information **121**.

At a time subsequent to the publishing of the second public key, a login request is received from the user when accessing the first device (**320**) and, responsive to the received request, the secret key is created using the second public key (**322**). As an example, the user **202** can log back into the first device **106a**. The login request **208a**, for example, can be received by the user login service **120**. At this time, the first device **106a** can also compute the secret key **3** using the first device's private key (e.g., **9**) and the public key (e.g., **8**) from the second different device **106b**. Device A calculations **500** shown in FIG. 2F provide example steps and formulas for computing the secret key.

The first anonymous identifier is associated with the secret key (**324**). For example, the first anonymous identifier (e.g., Device ID **2**) can be stored with hashed version of the secret key in the linked anonymous identifiers **122**. As a result, both anonymous identifiers are now linked. For example, the secret key, the first anonymous identifier, and the second different anonymous identifier are stored as an entry in a table, e.g., row **234**. In some implementations, the association maps the secret key to both the first and the second different anonymous identifiers. In some implementations, one or more associations can be removed (e.g., deleted from the linked anonymous identifiers **122**) after expiration of a first time period (e.g., 24 hours, 48 hours, or some other time period). In some implementations, the time period can be associated with an amount of time after which the user would have been expected to have logged out from either the first device or the second different device.

A request for content is received from either the first device including the first anonymous identifier or the second different device including the second different anonymous identifier (**326**). In one example, referring to FIG. 2E, the content management system **110** can receive, from the first device **106a**, the request for content **240a** that includes the anonymous identifier Device ID **1**. In another example, the content management system **110** can receive, from the second different device **106b**, the request for content **240b** that includes the anonymous identifier Device ID **2**.

Content is provided in response to the request using the association (**328**). For example, depending on which device sent the request for content **240a** or **240b**, the content management system **110** can provide content items **246a** or **246b** to either the first device **106a** or the second different device **106b**, respectively.

In some implementations, providing content in response to the request can further include identifying the user based on the association and providing content of interest to the user. For example, information (e.g., an interest in sports) that the user has provided in a user profile (or other information

20

provided by and/or known about the user) can be used to select content which is likely of interest to the user.

Some implementations of the process **300** can include steps for linking additional devices, e.g., a third device and/or additional devices. For example, a login request can be received from a third different device used by the user, the login request being associated with a third different anonymous identifier associated with the third different device. A third different public-private key pair can be created, including a third public key. The third private key can be stored locally on the third different device, and the third public key can be published (e.g., in the user login information **121**). A secret key can be created using one of either the first public key or the second public key, in addition to the third different device's private key, e.g., using steps and formulas shown in FIG. 2F. An association between the secret key, the first anonymous identifier, the second different anonymous identifier and the third different anonymous identifier can be stored, e.g., in the linked anonymous identifiers **122**. Subsequently, a request for content can be received from either the first device including the first anonymous identifier, the second different device including the second different anonymous identifier, or the third different device including the third different anonymous identifier. In response to request, content (e.g., content items **246a** or **246b**, or content items for the third different device) can be provided using the association.

FIG. 3B is a flowchart of an example process **340** for providing content to a user on any of multiple linked devices associated with the user. In some implementations, the content management system **110** and/or the user login service **120** can perform steps of the process **340** using instructions that are executed by one or more processors. FIGS. 1-2F are used to provide example structures for performing the steps of the process **340**.

Multiple anonymous identifiers associated with a user are linked by a service using a key exchange protocol without storing personally identifiable information associated with the user in the linking (**342**). For example, anonymous identifiers (e.g., browser cookies, or device Device IDs **1** and **2**) of the first device **106a** and the second different device **106b**, respectively, can be linked by the user login service **120**. The linking, for example, can occur using key exchange techniques described above, including using public, private and secret key calculations shown in FIG. 2E. In some implementations, public keys can be published on the user login service **120**, private keys can be stored on the corresponding local device, and secret keys can be stored in a third location (e.g., linked anonymous identifiers **122**). Other techniques can be used to link the devices, and more than two devices can be linked.

In some implementations, linking multiple anonymous identifiers can include receiving a login request (e.g., login requests **208a** or **208b**) from the user from plural different devices, determining a secret key using published public key information from another device associated with the user (where the secret key does not include any personally identifiable information associated with the user) and mapping the secret key to an anonymous identifier associated with each login request. For example, the secret key can be a secret key stored in the linked anonymous identifiers **122**, which does not include information about the user that can be traced back to the user (i.e., without having access to the information from the user login information **121**, the linked anonymous identifiers **122**, and private keys stored on the various user devices).

21

In some implementations, determining the secret key can include, at each device, creating a public-private key pair, publishing a public key of the public-private key pair, and using a private key of the public-private key pair and a public key of another device to compute the secret key.

Requests for content from a client device associated with the user are received at the service, where each request includes one of the anonymous identifiers (344). For example, referring to FIG. 2E, the content management system 110 can receive the request for content 240a that includes the anonymous identifier Device ID 1 corresponding to the first device 106a. In another example, the content management system 110 can receive the request for content 240b that includes the anonymous identifier Device ID 2 corresponding to the second different device 106b.

Content associated with the user is provided that is responsive to the received requests and based on the linking (346). For example, the content management system 110 can provide content items 246a or 246b to either the first device 106a or the second different device 106b, respectively, depending on which device sent the request for content 240a or 240b.

FIG. 3C is a flowchart of an example process 360 for providing content to a user on any of multiple devices linked using public-private keys. In some implementations, the content management system 110 and/or the user login service 120 can perform steps of the process 360 using instructions that are executed by one or more processors. FIGS. 1-2F are used to provide example structures for performing the steps of the process 360.

Public-private key pairs are created for a user each time the user logs into a service from a different device including publishing respective public keys of the user in a directory entry associated with the user (362). For example, FIGS. 2A-2D show a sequence of actions that use public-private key pairs to link the first device 106a and the second different device 106b. The public keys in this example are stored in the user login information 121.

A secret key is created by each device using a public key of another device that is stored in the directory (364). For example, FIGS. 2C-2D show a sequence of actions that determine the secret key for each of the first device 106a and the second different device 106b using the public key of the other device.

The secret keys are associated with a plurality of anonymous identifiers, each anonymous identifier assigned to the user during a session associated with a respective different device (366). As an example, the secret key is stored in the linked anonymous identifiers 122. Steps and formulas for computing the secret keys are shown in FIG. 2E.

Content is provided that is associated with the user and based at least in part on the association (368). For example, depending on which device sent the request for content 240a or 240b, the content management system 110 can provide content items 246a or 246b to either the first device 106a or the second different device 106b, respectively.

FIG. 3D is a flowchart of an example process 380 for limiting impressions of content based on frequency capping across multiple requesting sources. In some implementations, the content management system 110 and/or the user login service 120 can perform steps of the process 380 using instructions that are executed by one or more processors. FIGS. 1-2H are used to provide example structures/interfaces associated with the steps of the process 380.

Impressions of content to a user are identified, each impression including a time, where identifying impressions includes identifying impressions to the user accessing resources using different requesting sources (382). For example, the content

22

management system 110 can identify impressions 278a-278c of the Camera X advertisement to a user that is associated with multiple different user devices (or some other requesting sources).

Impression data for the identified impressions is stored in association with the user, including storing impression data on a per requesting source basis such that impressions that were received on a respective requesting source can be identified (384). For example, as impressions occur, the content management system 110 can update impression information 280 to identify the impressions 278a-278c that have occurred on various devices associated with the user 202. The information is stored in association with the anonymous identifiers 232a and 232b, which are associated with user 202.

Requesting source characteristic information that characterizes a given requesting source is stored in association with the impression data (386). For example, information stored with the anonymous identifiers 232a and 232b can include the type of requesting source, e.g., identifying the first device 106a as mobile and the second different device 106b as non-mobile. The mobile/non-mobile characteristic information is one example of information that is definable by the content sponsor for a campaign and stored in multi-device impression constraints 123. For example, the constraints stored for the Camera X campaign can designate the number and/or sequence of impressions of the Camera X advertisement that are to be presented to mobile and non-mobile devices.

Information is identified, including parameters that require limits on a number of impressions that are to occur in a time period for a particular type of requesting source (388). As an example, the content sponsor can use the content sponsor interface 256 to identify frequency capping constraints as described above with reference to FIG. 2G. The information can be stored in the multi-device impression constraints 123, e.g., identifying the number of impressions of the Camera X advertisement that are to be presented on various user devices that are linked and associated with the same user. The frequency capping constraints can also identify the characteristics of the requesting resources (e.g., mobile vs. non-mobile) that are permitted to receive the impressions over a time period T.

A request is received for content to be displayed on a particular requesting source associated with the user (390). For example, the content management system 110 can receive the request for content 240a from the first user device 106a or the request for content 240b from the second different user device 106a.

A determination is made regarding whether impressions available for that type of requesting source have been satisfied, based at least in part on the impression data and the parameters, and if not, enabling delivery of a content item associated with a campaign to the requesting source responsive to the received request (392). As an example, using impression information 280 and multi-device impression constraints 123, the content management system 110 can determine if of a content item (e.g., the Camera X advertisement) is to be presented to the requesting source (e.g., device 106a or 106b).

FIG. 4 is a block diagram of computing devices 400, 450 that may be used to implement the systems and methods described in this document, as either a client or as a server or plurality of servers. Computing device 400 is intended to represent various forms of digital computers, such as laptops, desktops, workstations, personal digital assistants, servers, blade servers, mainframes, and other appropriate computers. Computing device 400 is further intended to represent any other typically non-mobile devices, such as televisions or

other electronic devices with one or more processors embedded therein or attached thereto. Computing device 450 is intended to represent various forms of mobile devices, such as personal digital assistants, cellular telephones, smartphones, and other computing devices. The components shown here, their connections and relationships, and their functions, are meant to be exemplary only, and are not meant to limit implementations of the inventions described and/or claimed in this document.

Computing device 400 includes a processor 402, memory 404, a storage device 406, a high-speed interface 408 connecting to memory 404 and high-speed expansion ports 410, and a low speed interface 412 connecting to low speed bus 414 and storage device 406. Each of the components 402, 404, 406, 408, 410, and 412, are interconnected using various busses, and may be mounted on a common motherboard or in other manners as appropriate. The processor 402 can process instructions for execution within the computing device 400, including instructions stored in the memory 404 or on the storage device 406 to display graphical information for a GUI on an external input/output device, such as display 416 coupled to high speed interface 408. In other implementations, multiple processors and/or multiple buses may be used, as appropriate, along with multiple memories and types of memory. Also, multiple computing devices 400 may be connected, with each device providing portions of the necessary operations (e.g., as a server bank, a group of blade servers, or a multi-processor system).

The memory 404 stores information within the computing device 400. In one implementation, the memory 404 is a computer-readable medium. In one implementation, the memory 404 is a volatile memory unit or units. In another implementation, the memory 404 is a non-volatile memory unit or units.

The storage device 406 is capable of providing mass storage for the computing device 400. In one implementation, the storage device 406 is a computer-readable medium. In various different implementations, the storage device 406 may be a floppy disk device, a hard disk device, an optical disk device, or a tape device, a flash memory or other similar solid state memory device, or an array of devices, including devices in a storage area network or other configurations. In one implementation, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory 404, the storage device 406, or memory on processor 402.

The high speed controller 408 manages bandwidth-intensive operations for the computing device 400, while the low speed controller 412 manages lower bandwidth-intensive operations. Such allocation of duties is exemplary only. In one implementation, the high-speed controller 408 is coupled to memory 404, display 416 (e.g., through a graphics processor or accelerator), and to high-speed expansion ports 410, which may accept various expansion cards (not shown). In the implementation, low-speed controller 412 is coupled to storage device 406 and low-speed expansion port 414. The low-speed expansion port, which may include various communication ports (e.g., USB, Bluetooth, Ethernet, wireless Ethernet) may be coupled to one or more input/output devices, such as a keyboard, a pointing device, a scanner, or a networking device such as a switch or router, e.g., through a network adapter.

The computing device 400 may be implemented in a number of different forms, as shown in the figure. For example, it

may be implemented as a standard server 420, or multiple times in a group of such servers. It may also be implemented as part of a rack server system 424. In addition, it may be implemented in a personal computer such as a laptop computer 422. Alternatively, components from computing device 400 may be combined with other components in a mobile device (not shown), such as device 450. Each of such devices may contain one or more of computing device 400, 450, and an entire system may be made up of multiple computing devices 400, 450 communicating with each other.

Computing device 450 includes a processor 452, memory 464, an input/output device such as a display 454, a communication interface 466, and a transceiver 468, among other components. The device 450 may also be provided with a storage device, such as a microdrive or other device, to provide additional storage. Each of the components 450, 452, 464, 454, 466, and 468, are interconnected using various buses, and several of the components may be mounted on a common motherboard or in other manners as appropriate.

The processor 452 can process instructions for execution within the computing device 450, including instructions stored in the memory 464. The processor may also include separate analog and digital processors. The processor may provide, for example, for coordination of the other components of the device 450, such as control of user interfaces, applications run by device 450, and wireless communication by device 450.

Processor 452 may communicate with a user through control interface 458 and display interface 456 coupled to a display 454. The display 454 may be, for example, a TFT LCD display or an OLED display, or other appropriate display technology. The display interface 456 may comprise appropriate circuitry for driving the display 454 to present graphical and other information to a user. The control interface 458 may receive commands from a user and convert them for submission to the processor 452. In addition, an external interface 462 may be provided in communication with processor 452, so as to enable near area communication of device 450 with other devices. External interface 462 may provide, for example, for wired communication (e.g., via a docking procedure) or for wireless communication (e.g., via Bluetooth or other such technologies).

The memory 464 stores information within the computing device 450. In one implementation, the memory 464 is a computer-readable medium. In one implementation, the memory 464 is a volatile memory unit or units. In another implementation, the memory 464 is a non-volatile memory unit or units. Expansion memory 474 may also be provided and connected to device 450 through expansion interface 472, which may include, for example, a subscriber identification module (SIM) card interface. Such expansion memory 474 may provide extra storage space for device 450, or may also store applications or other information for device 450. Specifically, expansion memory 474 may include instructions to carry out or supplement the processes described above, and may include secure information also. Thus, for example, expansion memory 474 may be provided as a security module for device 450, and may be programmed with instructions that permit secure use of device 450. In addition, secure applications may be provided via the SIM cards, along with additional information, such as placing identifying information on the SIM card in a non-hackable manner.

The memory may include for example, flash memory and/or MRAM memory, as discussed below. In one implementation, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, perform one or more meth-

ods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory **464**, expansion memory **474**, or memory on processor **452**.

Device **450** may communicate wirelessly through communication interface **466**, which may include digital signal processing circuitry where necessary. Communication interface **466** may provide for communications under various modes or protocols, such as GSM voice calls, SMS, EMS, or MMS messaging, CDMA, TDMA, PDC, WCDMA, CDMA2000, or GPRS, among others. Such communication may occur, for example, through radio-frequency transceiver **468**. In addition, short-range communication may occur, such as using a Bluetooth, WiFi, or other such transceiver (not shown). In addition, GPS receiver module **470** may provide additional wireless data to device **450**, which may be used as appropriate by applications running on device **450**.

Device **450** may also communicate audibly using audio codec **460**, which may receive spoken information from a user and convert it to usable digital information. Audio codec **460** may likewise generate audible sound for a user, such as through a speaker, e.g., in a handset of device **450**. Such sound may include sound from voice telephone calls, may include recorded sound (e.g., voice messages, music files, etc.) and may also include sound generated by applications operating on device **450**.

The computing device **450** may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a cellular telephone **480**. It may also be implemented as part of a smartphone **482**, personal digital assistant, or other mobile device.

Various implementations of the systems and techniques described here can be realized in digital electronic circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms “machine-readable medium” “computer-readable medium” refers to any computer program product, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term “machine-readable signal” refers to any signal used to provide machine instructions and/or data to a programmable processor.

To provide for interaction with a user, the systems and techniques described here can be implemented on a computer having a display device (e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor) for displaying information to the user and a keyboard and a pointing device (e.g., a mouse or a trackball) by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback

(e.g., visual feedback, auditory feedback, or tactile feedback); and input from the user can be received in any form, including acoustic, speech, or tactile input.

The systems and techniques described here can be implemented in a computing system that includes a back end component (e.g., as a data server), or that includes a middleware component (e.g., an application server), or that includes a front end component (e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the systems and techniques described here), or any combination of such back end, middleware, or front end components. The components of the system can be interconnected by any form or medium of digital data communication (e.g., a communication network). Examples of communication networks include a local area network (“LAN”), a wide area network (“WAN”), and the Internet.

The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any inventions or of what may be claimed, but rather as descriptions of features specific to particular implementations of particular inventions. Certain features that are described in this specification in the context of separate implementations can also be implemented in combination in a single implementation. Conversely, various features that are described in the context of a single implementation can also be implemented in multiple implementations separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the implementations described above should not be understood as requiring such separation in all implementations, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

Thus, particular implementations of the subject matter have been described. Other implementations are within the scope of the following claims. In some cases, the actions recited in the claims can be performed in a different order and still achieve desirable results. In addition, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking and parallel processing may be advantageous.

What is claimed is:

1. A computer-implemented method for delivering content to a user, comprising:
 - identifying previous impressions of content to a user, each previous impression including a time that the particular impression was previously presented to the user and a

27

content item that was presented, wherein identifying impressions includes identifying impressions to the user accessing resources using two or more different requesting sources wherein the requesting sources are selected from the group comprising a device, a browser on a device, or an application on the device;

storing impression data for the identified impressions in association with the user including storing impression data on a per requesting source basis per content item such that specific impressions of a content item that were received on a respective requesting source can be identified;

storing requesting source characteristic information that characterizes a given requesting source in association with the impression data;

identifying information including parameters that require limits on a number of impressions that are to occur in a time period for a particular type of requesting source for a given content item wherein the information includes frequency information for frequency of display of the content item, and wherein identifying impressions and storing impression data further includes identifying a number of impressions of each content item the user;

receiving a request for content to be displayed on a particular requesting source associated with the user; and

determining, using one or more processors, when impressions available for that type of requesting source have been satisfied based at least in part on the impression data and the parameters, and when not, enabling delivery of a content item associated with a campaign to the requesting source responsive to the received request.

2. The method of claim 1 wherein the content item is an advertisement.

28

3. The method of claim 1 wherein identifying impressions includes counting a number of impressions of the content item that have been made on the various requesting sources.

4. The method of claim 1 wherein the information includes threshold limits on a per requesting source basis for impressions of the content item.

5. The method of claim 1 wherein storing impression data includes storing impression data in association with a cookie that is linked to a requesting source.

6. The method of claim 5 wherein cookies of requesting sources associated with a user are linked using a Diffie-Hellman key protocol.

7. The method of claim 6 wherein the cookies are linked using a secret key derived from a seed that is unique to the user.

8. The method of claim 5 further comprising:
storing a mapping of cookies associated with the user including storing the impression data for each cookie.

9. The method of claim 1 wherein the requesting source characteristic information characterizes a type of device, and wherein the type of device is selected from a group comprising a mobile device, a desktop device or a tablet.

10. The method of claim 1 wherein the information includes limits for two different requesting source types.

11. The method of claim 1 wherein determining impressions available includes comparing a number of impressions in the impression data for a given device type associated with the received request with a threshold number of impressions for the given device type as specified in the information for the campaign.

12. The method of claim 1 wherein determining when impressions are available includes determining a next content item in a sequence to deliver responsive to the request based at least in part on the stored impression data.

* * * * *